

Página 1

Página 2

Esta página foi intencionalmente deixada em branco

Página 3

REDES DE COMPUTADORES

QUINTA EDIÇÃO

Página 4

Esta página foi intencionalmente deixada em branco

Página 5

REDES DE COMPUTADORES

QUINTA EDIÇÃO

ANDREW S. TANENBAUM

Vrije Universiteit

Amsterdã, Holanda

DAVID J. WETHERALL

universidade de Washington

Seattle, WA

PRENTICE HALL

Boston Columbus Indianápolis Nova York São Francisco Upper Saddle River

Amsterdã Cidade do Cabo Dubai Londres Madri Milão Paris Montreal Toronto

Delhi Cidade do México São Paulo Sydney Hong Kong Seul Singapura Tapei Tóquio

Página 6

Diretora Editorial : Marcia Horton

Editor-chefe : Michael Hirsch

Editor Executivo : Tracy Dunkelberger

Editor Assistente : Melinda Haggerty

Assistente Editorial : Allison Michael

Vice-presidente de marketing : Patrice Jones

Gerente de marketing : Yezan Alayan

Coordenadora de marketing : Kathryn Ferranti

Vice-presidente de produção : Vince O'Brien

Editor Gerente : Jeff Holcomb

Supervisor Sênior de Operações : Alan Fischer

Compradora de manufatura : Lisa McDowell

Direção da capa : Andrew S. Tanenbaum,

David J. Wetherall, Tracy Dunkelberger

Diretor de Arte : Linda Knowles

Designer da capa : Susan Paradise

Ilustração da capa : Jason Consalvo

Design de interiores : Andrew S. Tanenbaum

Gerente de Projeto de Produção AV:

Gregory L. Dulles

Ilustrações internas: Laserwords, Inc.

Editor de mídia : Daniel Sandin

Composição : Andrew S. Tanenbaum

Editora de Texto : Rachel Head

Revisor : Joe Ruddick

Impressora / Fichário : Courier / Westford

Impressora de capa : Lehigh-Phoenix Color /

Hagerstown

Créditos e agradecimentos emprestados de outras fontes e reproduzidos, com permissão, neste livro aparecem na página apropriada dentro do texto.

Muitas das designações de fabricantes e vendedores para distinguir seus produtos são reivindicadas como marcas registradas. Onde essas designações aparecem neste livro, e a editora foi ciente de uma reivindicação de marca registrada, as designações foram impressas em maiúsculas iniciais ou todas em maiúsculas.

Copyright © 2011, 2003, 1996, 1989, 1981, Pearson Education, Inc., publicado como Prentice Corredor. Todos os direitos reservados. Fabricado nos Estados Unidos da América. Esta publicação é protegido por direitos autorais, e a permissão deve ser obtida do editor antes de qualquer reprodução proibida, armazenamento em um sistema de recuperação ou transmissão em qualquer forma ou por qualquer meios, eletrônico, mecânico, fotocópia, gravação, ou similar. Para obter permissão (s) para usar o material deste trabalho, envie uma solicitação por escrito à Pearson Education, Inc., Departamento de Permissões, 501 Boylston Street, Suite 900, Boston, Massachusetts 02116.

Dados de Catalogação na Publicação da Biblioteca do Congresso

Tanenbaum, Andrew S., 1944-

Redes de computadores / Andrew S. Tanenbaum, David J. Wetherall. - 5^a ed.

p. cm.

Inclui referências bibliográficas e índice.

ISBN-13: 978-0-13-212695-3 (papel alcalino)

ISBN-10: 0-13-212695-8 (papel alcalino)

1. Redes de computadores. I. Wetherall, D. (David) II. Título.

TK5105.5.T36 2011

004.6 - dc22

2010034366

10 9 8 7 6 5 4 3 2 1 — CRW — 14 13 12 11 10

Página 7

*Para Suzanne, Barbara, Daniel, Aron, Marvin, Matilde,
e a memória de Bram e Sweetie π (AST)*

Para Katrin, Lucy e Pepper (DJW)

Página 8

Esta página foi intencionalmente deixada em branco

Página 9

CONTEÚDO

PREFÁCIO

xix

1 INTRODUÇÃO

1

1.1 USOS DE REDES DE COMPUTADORES, 3

1.1.1 Aplicativos de negócios, 3

1.1.2 Aplicativos domésticos, 6

1.1.3 Usuários móveis, 10

1.1.4 Questões Sociais, 14

1.2 HARDWARE DE REDE, 17

1.2.1 Redes de área pessoal, 18

1.2.2 Redes locais, 19

1.2.3 Redes de Área Metropolitana, 23

1.2.4 Redes de longa distância, 23

1.2.5 Internetworks, 28

1.3 SOFTWARE DE REDE, 29

1.3.1 Hierarquias de protocolo, 29

1.3.2 Problemas de projeto para as camadas, 33

1.3.3 Serviço Orientado a Conexão Versus Sem Conexão, 35
1.3.4 Primitivas de serviço, 38
1.3.5 A relação dos serviços com os protocolos, 40
1.4 MODELOS DE REFERÊNCIA, 41
1.4.1 O Modelo de Referência OSI, 41
1.4.2 O Modelo de Referência TCP / IP, 45
1.4.3 O modelo usado neste livro, 48

vii

Página 10

viii

CONTEÚDO

1.4.4 Uma comparação dos modelos de referência OSI e TCP / IP *, 49
1.4.5 Uma crítica do modelo OSI e protocolos *, 51
1.4.6 Uma crítica do modelo de referência TCP / IP *, 53
1.5 EXEMPLO DE REDES, 54
1.5.1 A Internet, 54
1.5.2 Redes de telefonia móvel de terceira geração *, 65
1.5.3 LANs sem fio: 802.11 *, 70
1.5.4 RFID e Redes de Sensores *, 73
1.6 PADRONIZAÇÃO DA REDE *, 75
1.6.1 Quem é quem no mundo das telecomunicações, 77
1.6.2 Quem é quem no mundo dos padrões internacionais, 78
1.6.3 Quem é quem no mundo dos padrões da Internet, 80
1.7 UNIDADES MÉTRICAS, 82
1.8 ESBOÇO DO RESTO DO LIVRO, 83
1.9 RESUMO, 84

2 A CAMADA FÍSICA

89

2.1 A BASE TEÓRICA PARA COMUNICAÇÃO DE DADOS, 90
2.1.1 Análise de Fourier, 90
2.1.2 Sinais de largura de banda limitada, 90
2.1.3 A taxa máxima de dados de um canal, 94
2.2 MEIOS DE TRANSMISSÃO GUIADA, 95
2.2.1 Mídia magnética, 95
2.2.2 Pares Trançados, 96
2.2.3 Cabo coaxial, 97
2.2.4 Linhas de energia, 98
2.2.5 Fibra Óptica, 99
2.3 TRANSMISSÃO SEM FIO, 105
2.3.1 O Espectro Eletromagnético, 105
2.3.2 Transmissão de Rádio, 109
2.3.3 Transmissão de Microondas, 110
2.3.4 Transmissão infravermelha, 114
2.3.5 Transmissão de luz, 114

Página 11

CONTEÚDO

ix

2.4 SATÉLITES DE COMUNICAÇÃO *, 116
2.4.1 Satélites geoestacionários, 117
2.4.2 Satélites de órbita terrestre média, 121
2.4.3 Satélites de órbita terrestre baixa, 121
2.4.4 Satélites Versus Fibra, 123
2.5 MODULAÇÃO DIGITAL E MULTIPLEXAGEM, 125

2.5.1 Transmissão de banda base, 125
2.5.2 Transmissão de banda passante, 130
2.5.3 Multiplexação por Divisão de Freqüência, 132
2.5.4 Multiplexação por Divisão de Tempo, 135
2.5.5 Multiplexação por divisão de código, 135
2.6 A REDE TELEFÔNICA PÚBLICA COMUTADA, 138
2.6.1 Estrutura do Sistema Telefônico, 139
2.6.2 A Política dos Telefones, 142
2.6.3 O Loop Local: Modems, ADSL e Fibra, 144
2.6.4 Troncos e Multiplexação, 152
2.6.5 Comutação, 161
2.7 O SISTEMA DE TELEFONE MÓVEL *, 164
2.7.1 Telefones celulares de primeira geração (coco1G): voz analógica, 166
2.7.2 Telefones celulares de segunda geração (2G): Voz digital, 170
2.7.3 Telefones móveis de terceira geração (3G): voz e dados digitais, 174
2.8 TELEVISÃO A CABO *, 179
2.8.1 Antena de Televisão Comunitária, 179
2.8.2 Internet por cabo, 180
2.8.3 Alociação de espectro, 182
2.8.4 Modems a cabo, 183
2.8.5 ADSL Versus Cabo, 185
2.9 RESUMO, 186

3 A CAMADA DE LINK DE DADOS

193

3.1 QUESTÕES DE PROJETO DA CAMADA DE LINK DE DADOS, 194
3.1.1 Serviços prestados à camada de rede, 194
3.1.2 Enquadramento, 197
3.1.3 Controle de Erro, 200
3.1.4 Controle de fluxo, 201

Página 12

X

CONTEÚDO

3.2 DETEÇÃO E CORREÇÃO DE ERROS, 202
3.2.1 Códigos de correção de erros, 204
3.2.2 Códigos de detecção de erros, 209
3.3 PROTOCOLOS DE LINK DE DADOS ELEMENTARES, 215
3.3.1 Um protocolo simplex utópico, 220
3.3.2 Um Protocolo Simplex Stop-and-Wait para um canal livre de erros, 221
3.3.3 Um Protocolo Simplex Stop-and-Wait para um canal ruidoso, 222
3.4 PROTOCOLOS DE JANELA DE DESLIZAMENTO, 226
3.4.1 Um protocolo de janela deslizante de um bit, 229
3.4.2 Um protocolo usando Go-Back-N, 232
3.4.3 Um protocolo usando repetição seletiva, 239
3.5 PROTOCOLOS DE EXEMPLO DE LINK DE DADOS, 244
3.5.1 Pacote sobre SONET, 245
3.5.2 ADSL (Asymmetric Digital Subscriber Loop), 248
3.6 RESUMO, 251

4 A SUBCAMADA DE CONTROLE DE MÉDIO ACESSO 257

4.1 O PROBLEMA DE ALOCAÇÃO DE CANAIS, 258
4.1.1 Alocação de canal estático, 258
4.1.2 Premissas para alocação dinâmica de canais, 260
4.2 PROTOCOLOS DE ACESSO MÚLTIPLOS, 261
4.2.1 ALOHA, 262
4.2.2 Protocolos de Acesso Múltiplo do Carrier Sense, 266

4.2.3 Protocolos livres de colisão, 269
4.2.4 Protocolos de Contenção Limitada, 274
4.2.5 Protocolos de LAN sem fio, 277
4.3 ETHERNET, 280
4.3.1 Camada Física Ethernet Clássica, 281
4.3.2 Protocolo Subcamada MAC Ethernet Clássico, 282
4.3.3 Desempenho Ethernet, 286
4.3.4 Ethernet Comutada, 288

Página 13

CONTEÚDO

XI

4.3.5 Fast Ethernet, 290
4.3.6 Gigabit Ethernet, 293
4.3.7 Ethernet 10-Gigabit, 296
4.3.8 Retrospectiva na Ethernet, 298
4.4 LANS SEM FIO, 299
4.4.1 A arquitetura 802.11 e pilha de protocolo, 299
4.4.2 A camada física 802.11, 301
4.4.3 O protocolo de subcamada 802.11 MAC, 303
4.4.4 A Estrutura do Frame 802.11, 309
4.4.5 Serviços, 311
4.5 BANDA LARGA WIRELESS *, 312
4.5.1 Comparação de 802.16 com 802.11 e 3G, 313
4.5.2 A Arquitetura e Pilha de Protocolo 802.16, 314
4.5.3 A Camada Física 802.16, 316
4.5.4 O protocolo de subcamada 802.16 MAC, 317
4.5.5 A Estrutura do Quadro 802.16, 319
4.6 BLUETOOTH *, 320
4.6.1 Arquitetura Bluetooth, 320
4.6.2 Aplicativos Bluetooth, 321
4.6.3 A pilha de protocolo Bluetooth, 322
4.6.4 A Camada de Rádio Bluetooth, 324
4.6.5 As camadas de link Bluetooth, 324
4.6.6 A Estrutura do Frame Bluetooth, 325
4.7 RFID *, 327
4.7.1 Arquitetura EPC Gen 2, 327
4.7.2 Camada Física EPC Gen 2, 328
4.7.3 Camada de Identificação de Tag EPC Gen 2, 329
4.7.4 Formatos de mensagem de identificação de tag, 331
4.8 COMUTAÇÃO DA CAMADA DE LINK DE DADOS, 332
4.8.1 Usos de Pontes, 332
4.8.2 Pontes de Aprendizagem, 334
4.8.3 Spanning Tree Bridges, 337
4.8.4 Repetidores, Hubs, Pontes, Switches, Roteadores e Gateways, 340
4.8.5 LANs virtuais, 342
4.9 RESUMO, 349

Página 14

xii

CONTEÚDO

5 A CAMADA DE REDE
355

5.1 PROBLEMAS DE PROJETO DA CAMADA DE REDE, 355
5.1.1 Comutação de pacotes de armazenamento e encaminhamento, 356

5.1.2 Serviços prestados à camada de transporte, 356
5.1.3 Implementação do Serviço sem Conexão, 358
5.1.4 Implementação de Serviço Orientado a Conexão, 359
5.1.5 Comparação de Redes de Circuito Virtual e Datagrama, 361
5.2 ALGORITMOS DE ROTEAMENTO, 362
5.2.1 O Princípio da Otimidade, 364
5.2.2 Algoritmo de caminho mais curto, 366
5.2.3 Inundações, 368
5.2.4 Roteamento de vetor de distância, 370
5.2.5 Roteamento de Estado de Link, 373
5.2.6 Roteamento Hierárquico, 378
5.2.7 Roteamento de difusão, 380
5.2.8 Roteamento Multicast, 382
5.2.9 Roteamento Anycast, 385
5.2.10 Roteamento para hosts móveis, 386
5.2.11 Roteamento em Redes Ad Hoc, 389
5.3 ALGORITMOS DE CONTROLE DE CONGESTÃO, 392
5.3.1 Abordagens para Controle de Congestionamento, 394
5.3.2 Roteamento sensível ao tráfego, 395
5.3.3 Controle de Admissão, 397
5.3.4 Estrangulamento de tráfego, 398
5.3.5 Descarte de Carga, 401
5.4 QUALIDADE DE SERVIÇO, 404
5.4.1 Requisitos de Aplicação, 405
5.4.2 Traffic Shaping, 407
5.4.3 Programação de pacotes, 411
5.4.4 Controle de Admissão, 415
5.4.5 Serviços Integrados, 418
5.4.6 Serviços Diferenciados, 421
5.5 INTERNETWORKING, 424
5.5.1 Como as redes diferem, 425
5.5.2 Como as redes podem ser conectadas, 426
5.5.3 Tunelamento, 429

Página 15

CONTEÚDO

xiii

5.5.4 Roteamento Internetwork, 431
5.5.5 Fragmentação de Pacote, 432
5.6 A CAMADA DE REDE NA INTERNET, 436
5.6.1 O Protocolo IP Versão 4, 439
5.6.2 Endereços IP, 442
5.6.3 IP Versão 6, 455
5.6.4 Protocolos de Controle da Internet, 465
5.6.5 Comutação de etiqueta e MPLS, 470
5.6.6 OSPF - Um protocolo de roteamento de gateway interior, 474
5.6.7 BGP - O Protocolo de Roteamento de Gateway Exterior, 479
5.6.8 Multicast da Internet, 484
5.6.9 IP móvel, 485
5.7 RESUMO, 488

6 A CAMADA DE TRANSPORTE

495

6.1 O SERVIÇO DE TRANSPORTE, 495
6.1.1 Serviços prestados às camadas superiores, 496
6.1.2 Primitivas de serviço de transporte, 498
6.1.3 Soquetes Berkeley, 500

6.1.4 Um exemplo de programação de soquete: um servidor de arquivos da Internet, 503
6.2 ELEMENTOS DE PROTOCOLOS DE TRANSPORTE, 507
6.2.1 Enderecamento, 509
6.2.2 Estabelecimento de conexão, 512
6.2.3 Liberação de Conexão, 517
6.2.4 Controle de Erro e Controle de Fluxo, 522
6.2.5 Multiplexação, 527
6.2.6 Crash Recovery, 527
6.3 CONTROLE DE CONGESTÃO, 530
6.3.1 Alocação de largura de banda desejável, 531
6.3.2 Regulando a Taxa de Envio, 535
6.3.3 Problemas sem fio, 539
6.4 OS PROTOCOLOS DE TRANSPORTE DA INTERNET: UDP, 541
6.4.1 Introdução ao UDP, 541
6.4.2 Chamada de Procedimento Remoto, 543
6.4.3 Protocolos de Transporte em Tempo Real, 546

Página 16

xiv

CONTEÚDO

6.5 OS PROTOCOLOS DE TRANSPORTE DA INTERNET: TCP, 552
6.5.1 Introdução ao TCP, 552
6.5.2 O Modelo de Serviço TCP, 553
6.5.3 O Protocolo TCP, 556
6.5.4 O cabeçalho do segmento TCP, 557
6.5.5 Estabelecimento de Conexão TCP, 560
6.5.6 Liberação de Conexão TCP, 562
6.5.7 Modelagem de Gerenciamento de Conexão TCP, 562
6.5.8 Janela Deslizante TCP, 565
6.5.9 Gerenciamento de temporizador TCP, 568
6.5.10 Controle de Congestionamento TCP, 571
6.5.11 O Futuro do TCP, 581
6.6 QUESTÕES DE DESEMPENHO *, 582
6.6.1 Problemas de desempenho em redes de computadores, 583
6.6.2 Medição de Desempenho da Rede, 584
6.6.3 Design de host para redes rápidas, 586
6.6.4 Processamento de Segmento Rápido, 590
6.6.5 Compressão de cabeçalho, 593
6.6.6 Protocolos para Long Fat Networks, 595
6.7 REDE TOLERANTE DE ATRASO *, 599
6.7.1 Arquitetura DTN, 600
6.7.2 O Protocolo Bundle, 603
6.8 RESUMO, 605

7 A CAMADA DE APLICAÇÃO

611

7.1 DNS - O SISTEMA DE NOME DE DOMÍNIO, 611
7.1.1 O Espaço de Nomes DNS, 612
7.1.2 Registros de Recursos de Domínio, 616
7.1.3 Servidores de Nomes, 619
7.2 CORREIO ELETRÔNICO *, 623
7.2.1 Arquitetura e Serviços, 624
7.2.2 O Agente do Usuário, 626
7.2.3 Formatos de Mensagem, 630
7.2.4 Transferência de Mensagens, 637
7.2.5 Entrega Final, 643

CONTEÚDO

xv

- [7.3 WEB NO MUNDO INTEIRO, 646](#)
 - [7.3.1 Visão geral da arquitetura, 647](#)
 - [7.3.2 Páginas da Web estáticas, 662](#)
 - [7.3.3 Páginas da Web e Aplicativos da Web Dinâmicos, 672](#)
 - [7.3.4 HTTP - O protocolo de transferência de hipertexto, 683](#)
 - [7.3.5 A Web Móvel, 693](#)
 - [7.3.6 Pesquisa na web, 695](#)
- [7.4 TRANSMISSÃO DE ÁUDIO E VÍDEO, 697](#)
 - [7.4.1 Áudio Digital, 699](#)
 - [7.4.2 Vídeo Digital, 704](#)
 - [7.4.3 Streaming de mídia armazenada, 713](#)
 - [7.4.4 Streaming Live Media, 721](#)
 - [7.4.5 Conferência em Tempo Real, 724](#)
- [7.5 ENTREGA DE CONTEÚDO, 734](#)
 - [7.5.1 Conteúdo e tráfego da Internet, 736](#)
 - [7.5.2 Farms de Servidores e Proxies da Web, 738](#)
 - [7.5.3 Redes de distribuição de conteúdo, 743](#)
 - [7.5.4 Redes Peer-to-Peer, 748](#)
- [7.6 RESUMO, 757](#)

8 SEGURANÇA DA REDE

763

- [8.1 CRIPTOGRAFIA, 766](#)
 - [8.1.1 Introdução à criptografia, 767](#)
 - [8.1.2 Cifras de Substituição, 769](#)
 - [8.1.3 Cifras de transposição, 771](#)
 - [8.1.4 Pads de uso único, 772](#)
 - [8.1.5 Dois princípios criptográficos fundamentais, 776](#)
- [8.2 ALGORITMOS DE CHAVE SIMÉTRICA, 778](#)
 - [8.2.1 DES - The Data Encryption Standard, 780](#)
 - [8.2.2 AES - The Advanced Encryption Standard, 783](#)
 - [8.2.3 Modos de codificação, 787](#)
 - [8.2.4 Outras Cifras, 792](#)
 - [8.2.5 Criptoanálise, 792](#)

xvi

CONTEÚDO

- [8.3 ALGORITMOS DE CHAVE PÚBLICA, 793](#)
 - [8.3.1 RSA, 794](#)
 - [8.3.2 Outros Algoritmos de Chave Pública, 796](#)
- [8.4 ASSINATURAS DIGITAIS, 797](#)
 - [8.4.1 Assinaturas de chave simétrica, 798](#)
 - [8.4.2 Assinaturas de chave pública, 799](#)
 - [8.4.3 Resumos de mensagens, 800](#)
 - [8.4.4 O Ataque de Aniversário, 804](#)
- [8.5 GESTÃO DE CHAVES PÚBLICAS, 806](#)
 - [8.5.1 Certificados, 807](#)
 - [8.5.2 X.509, 809](#)
 - [8.5.3 Infraestruturas de chave pública, 810](#)
- [8.6 SEGURANÇA DE COMUNICAÇÃO, 813](#)
 - [8.6.1 IPsec, 814](#)

8.6.2 Firewalls, 818
8.6.3 Redes Privadas Virtuais, 821
8.6.4 Segurança sem fio, 822
8.7 PROTOCOLOS DE AUTENTICAÇÃO, 827
8.7.1 Autenticação baseada em uma chave secreta compartilhada, 828
8.7.2 Estabelecendo uma chave compartilhada: The Diffie-Hellman Key Exchange, 833
8.7.3 Autenticação usando um centro de distribuição de chaves, 835
8.7.4 Autenticação usando Kerberos, 838
8.7.5 Autenticação usando criptografia de chave pública, 840
8.8 SEGURANÇA DE E-MAIL *, 841
8.8.1 PGP - Pretty Good Privacy, 842
8.8.2 S / MIME, 846
8.9 SEGURANÇA DA WEB, 846
8.9.1 Ameaças, 847
8.9.2 Nomenclatura segura, 848
8.9.3 SSL - The Secure Sockets Layer, 853
8.9.4 Segurança de código móvel, 857
8.10 QUESTÕES SOCIAIS, 860
8.10.1 Privacidade, 860
8.10.2 Liberdade de expressão, 863
8.10.3 Copyright, 867
8.11 SUMÁRIO, 869

Página 19

CONTEÚDO

xvii

9 LISTA DE LEITURA E BIBLIOGRAFIA

877

9.1 SUGESTÕES PARA LEITURA ADICIONAL *, 877

9.1.1 Introdução e Obras Gerais, 878
--

9.1.2 A Camada Física, 879
--

9.1.3 A camada de enlace de dados, 880
--

9.1.4 Subcamada de controle de acesso ao meio, 880
--

9.1.5 A Camada de Rede, 881

9.1.6 A Camada de Transporte, 882

9.1.7 A Camada de Aplicativo, 882

9.1.8 Segurança de Rede, 883
--

9.2 BIBLIOGRAFIA ALFABÉTICA *, 884
--

ÍNDICE

905

Página 20

Esta página foi intencionalmente deixada em branco

Página 21

PREFÁCIO

Este livro está agora em sua quinta edição. Cada edição correspondeu a um diferente fase diferente na forma como as redes de computadores foram usadas. Quando a primeira edição surgiu em 1980, as redes eram uma curiosidade acadêmica. Quando a segunda edição surgiu em 1988, as redes eram utilizadas por universidades e grandes empresas. Quando a terceira edição apareceu em 1996, as redes de computadores, especialmente a Internet, tiveram tornou-se uma realidade diária para milhões de pessoas. Na quarta edição, em 2003, o fio-

menos redes e computadores móveis tornaram-se comuns para acessar o Web e a Internet. Agora, na quinta edição, as redes são sobre distribuição (especialmente vídeos usando CDNs e redes ponto a ponto) e dispositivos móveis telefones são pequenos computadores na Internet.

Novo na quinta edição

Entre as muitas mudanças neste livro, a mais importante é a adição do Prof. David J. Wetherall como co-autor. David traz um rico histórico em rede trabalhando, tendo iniciado o projeto de redes da área metropolitana de mais de 20 anos atrás. Ele trabalhou com a Internet e redes sem fio desde então e é professor da Universidade de Washington, onde lecionou e fazendo pesquisas sobre redes de computadores e tópicos relacionados na última década. Claro, o livro também tem muitas mudanças para acompanhar: em constante mudança mundo das redes de computadores. Entre eles estão revisados e novos materiais sobre Redes sem fio (802.12 e 802.16)

As redes 3G usadas por smartphones

RFID e redes de sensores

Distribuição de conteúdo usando CDNs

Redes ponto a ponto

Mídia em tempo real (de fontes armazenadas, streaming e ao vivo)

Telefonia pela Internet (voz sobre IP)

Redes tolerantes a atrasos

Segue uma lista mais detalhada de capítulo por capítulo.

xix

Página 22

XX

PREFÁCIO

O Capítulo 1 tem a mesma função introdutória da quarta edição, mas o conteúdo foi revisado e atualizado. Internet, telefone celular redes, 802.11 e RFID e redes de sensores são discutidos como exemplos de redes de computadores. Material na Ethernet original - com suas torneiras de vampiro - foi removido, junto com o material no caixa eletrônico.

O Capítulo 2, que cobre a camada física, expandiu a cobertura do digital modulação (incluindo OFDM, amplamente utilizado em redes sem fio) e rede 3G funciona (baseado em CDMA). Novas tecnologias são discutidas, incluindo fibra para a Rede doméstica e de linha de energia.

O Capítulo 3, sobre links ponto a ponto, foi aprimorado de duas maneiras. O material em códigos para detecção e correção de erros foi atualizado e também inclui uma breve descrição dos códigos modernos que são importantes na prática (por exemplo, convocódigos oficiais e LDPC). Os exemplos de protocolos agora usam Packet sobre SONET e ADSL. Infelizmente, o material de verificação do protocolo foi removido pois é pouco usado.

No Capítulo 4, na subcamada MAC, os princípios são atemporais, mas a tecnologia as tecnologias mudaram. Seções nas redes de exemplo foram refeitas consequentemente, incluindo Gigabit Ethernet, 802.11, 802.16, Bluetooth e RFID. Também atualizada é a cobertura de comutação de LAN, incluindo VLANs.

O Capítulo 5, na camada de rede, cobre o mesmo terreno que na quarta edição. As revisões foram para atualizar o material e adicionar profundidade, especialmente para qualidade de serviço (relevante para mídia em tempo real) e internetworking. As seções sobre BGP, OSPF e CIDR foram expandidas, assim como o tratamento de roteamento multicast. O roteamento anycast agora está incluído.

O Capítulo 6, sobre a camada de transporte, teve material adicionado, revisado e remudou-se. Novo material descreve redes tolerantes a atrasos e controle de congestionamento em geral. O material revisado atualiza e expande a cobertura do TCP com controle de gestação. O material removido descreveu a configuração da rede orientada à conexões, algo raramente visto mais.

O capítulo 7, sobre aplicativos, também foi atualizado e ampliado. Enquanto mater-

ial no DNS e no e-mail é semelhante ao da quarta edição, nos últimos anos houve muitos desenvolvimentos no uso da Web, streaming de mídia e entrega de conteúdo. Assim, as seções na Web e mídia de streaming têm sido atualizada. Uma nova seção cobre a distribuição de conteúdo, incluindo CDNs e redes ponto a ponto.

O Capítulo 8, sobre segurança, ainda abrange criptografia simétrica e de chave pública gráfica para confidencialidade e autenticidade. Material sobre as técnicas utilizadas em prática, incluindo firewalls e VPNs, foi atualizada, com novo material sobre Segurança 802.11 e Kerberos V5 adicionados.

Capítulo 9 contém uma lista renovada de leituras sugeridas e um abrangente bibliografia de mais de 300 citações da literatura atual. Mais da metade de estes são para artigos e livros escritos em 2000 ou mais tarde, e o resto são citações para papéis clássicos.

PREFÁCIO

xxi

Lista de acrônimos

Os livros de informática estão cheios de siglas. Este não é exceção. Quando chegar a hora você terminou de ler este, o seguinte deve soar um sino: ADSL, AES, AJAX, AODV, AP, ARP, ARQ, AS, BGP, BOC, CDMA, CDN, CGI, CIDR, CRL, CSMA, CSS, DCT, DES, DHCP, DHT, DIFS, DMCA, DMT, DMZ, DNS, DOCSIS, DOM, DSLAM, DTN, FCFS, FDD, FDDI, FDM, FEC, FIFO, FSK, FTP, GPRS, GSM, HDTV, HFC, HMAC, HTTP, IAB, ICANN, ICMP, IDEA, IETF, IMAP, IMP, IP, IPTV, IRTF, ISO, ISP, ITU, JPEG, JSP, JVM, LAN, LATA, LEC, LEO, LLC, LSR, LTE, MAN, MFJ, MIME, MPEG, MPLS, MSC, MTSO, MTU, NAP, NAT, NRZ, NSAP, OFDM, OSI, OSPF, PAWS, PCM, PGP, PIM, PKI, POP, POTS, PPP, PSTN, QAM, QPSK, RED, RFC, RFID, RPC, RSA, RTSP, SHA, SIP, SMTP, SNR, SOAP, SONET, SPE, SSL, TCP, TDD, TDM, TSAP, UDP, UMTS, URL, VLAN, VSAT, WAN, WDM e XML. Mas não

preocupação. Cada um aparecerá em **negrito** e será cuidadosamente definido antes de ser usava. Como um teste divertido, veja quantos você consegue identificar *antes de* ler o livro, escreva o número na margem e tente novamente *após* ler o livro.

Como usar o livro

Para ajudar os instrutores a usar este livro como um texto para cursos variando de trimestres a semestres, estruturamos os capítulos em principais e opcionais materiais. As seções marcadas com " * " no índice são opcionais uns. Se uma seção principal (por exemplo, 2.7) estiver marcada, todas as suas subseções serão opcionais.

Eles fornecem material sobre tecnologias de rede que é útil, mas pode ser omitido de um curso curto sem perda de continuidade. Claro, os alunos devem ser encorajados a ler essas seções também, na medida em que têm tempo, como todos os material está atualizado e de valor.

Materiais de recursos para instrutores

Os seguintes materiais de recursos de instrutores protegidos estão disponíveis no site da editora em www.pearsonhighered.com/tanenbaum. Para um nome de usuário e senha, entre em contato com o representante local da Pearson.

Manual de soluções

Slides de palestra em PowerPoint

Materiais de recursos dos alunos

Os recursos para os alunos estão disponíveis na Web Companion de acesso aberto link do site em www.pearsonhighered.com/tanenbaum, Incluindo Recursos da web, links para tutoriais, organizações, perguntas frequentes e muito mais Figuras, tabelas e programas do livro

Demonstração de esteganografia

Simuladores de protocolo

xxii

PREFÁCIO

Reconhecimentos

Muitas pessoas nos ajudaram ao longo da quinta edição. Gostaríamos de especialmente gostaria de agradecer a Emmanuel Agu (Worcester Polytechnic Institute), Yoris Au (Universidade do Texas em Antonio), Nikhil Bhargava (Aircom International, Inc.), Michael Buettner (Universidade de Washington), John Day (Universidade de Boston), Kevin Fall (Intel Labs), Ronald Fulle (Rochester Institute of Technology), Ben Greenstein (Intel Labs), Daniel Halperin (Universidade de Washington), Bob Kinicki (Worcester Polytechnic Institute), Tadayoshi Kohno (University of Washington), Sarvish Kulkarni (Universidade Villanova), Hank Levy (Universidade de Washington), Ratul Mahajan (Microsoft Research), Craig Partridge (BBN), Michael Piatek (Universidade de Washington), Joshua Smith (Intel Labs), Neil Spring (Universidade de Maryland), David Teneyuca (Universidade do Texas em Antonio), Tammy VanDe-grift (Universidade de Portland) e Bo Yuan (Instituto de Tecnologia de Rochester), para fornecer ideias e feedback. Melody Kadenko e Julie Svendsen forneceram suporte administrativo ao David.

Shivakant Mishra (Universidade do Colorado em Boulder) e Paul Nagin (Chimborazo Publishing, Inc.) pensou em muitos novos e desafiadores finais de capítulo problemas. Nossa editora na Pearson, Tracy Dunkelberger, foi sempre prestativa em muitos aspectos, grandes e pequenos. Melinda Haggerty e Jeff Holcomb fizeram um bom trabalho de manter as coisas funcionando perfeitamente. Steve Armstrong (LeTourneau Universidade) preparou os slides do PowerPoint. Stephen Turner (Universidade de Michigan em Flint) revisou habilmente os recursos da Web e os simuladores que acompanham o texto. Nossa revisora, Rachel Head, é uma estranha híbrida: ela tem olhos de águia e a memória de um elefante. Depois de ler todas as correções dela, nós dois não-descobrimos como passamos da terceira série.

Finalmente, chegamos às pessoas mais importantes. Suzanne passou por isso 19 vezes agora e ainda tem paciência e amor infinitos. Barbara e Marvin agora sabe a diferença entre bons e maus livros e são sempre um inspiração para produzir bons. Daniel e Matilde são acréscimos bem-vindos ao nossa família. É improvável que Aron leia este livro em breve, mas ele gosta das belas fotos na página 866 (AST). Katrin e Lucy forneceram apoio infinito e sempre man-envelheceu para manter um sorriso no rosto. Obrigado (DJW).

Um ndrew S . T ANENBAUM

D AVID J . W ETHERALL

1

INTRODUÇÃO

Cada um dos últimos três séculos foi dominado por uma única nova tecnologia. O século 18 foi a era dos grandes sistemas mecânicos que acompanharam o Revolução Industrial. O século 19 foi a era da máquina a vapor. Durante século 20, a tecnologia-chave foi a coleta, processamento e distribuição. Entre outros desenvolvimentos, vimos a instalação de redes telefônicas, a invenção do rádio e da televisão, o nascimento e o não crescimento cedido da indústria de informática, o lançamento do satélite de comunicação

Lite e, claro, a Internet.

Como resultado do rápido progresso tecnológico, essas áreas estão convergindo rapidamente no século 21 e as diferenças entre coletar, transportar, armazenar, e as informações de processamento estão desaparecendo rapidamente. Organizações com hun-drenos de escritórios espalhados por uma ampla área geográfica normalmente esperam ser capazes de

para examinar o status atual até mesmo de seu posto avançado mais remoto com o impulso de um botão. À medida que nossa capacidade de reunir, processar e distribuir informações aumenta, a demanda por um processamento de informações cada vez mais sofisticado cresce ainda mais rápido.

Embora a indústria de computadores ainda seja jovem em comparação com outras indústrias (por exemplo, automóveis e transporte aéreo), os computadores tornaram gress em pouco tempo. Durante as primeiras duas décadas de sua existência, os sistemas eram altamente centralizados, geralmente em uma única sala grande. Não infre-frequentemente, esta sala tinha paredes de vidro, através das quais os visitantes podiam admirar o grande maravilha eletrônica dentro. Uma empresa ou universidade de médio porte pode ter tido

1

Página 26

2

INTRODUÇÃO

INDIVÍDUO. 1

um ou dois computadores, enquanto instituições muito grandes tinham no máximo algumas dezenas. o

ideia de que dentro de quarenta anos, computadores muito mais poderosos, menores do que os de correio

os selos seriam produzidos em massa aos bilhões era pura ficção científica.

A fusão de computadores e comunicações teve uma influência profunda na forma como os sistemas de computador são organizados. O conceito outrora dominante do " centro de informática " como uma sala com um grande computador para o qual os usuários trazem seus

trabalho para processamento está agora totalmente obsoleto (embora os centros de dados contenham mil

areias de servidores da Internet estão se tornando comuns). O antigo modelo de uma única empresa computador que atende a todas as necessidades computacionais da organização foi substituído por um no qual um grande número de computadores separados, mas interconectados, fazem o trabalho. Esses sistemas são chamados de **redes de computadores**. O design e organização de essas redes são o assunto deste livro.

Ao longo do livro, usaremos o termo " rede de computadores " para significar uma coluna seleção de computadores autônomos interconectados por uma única tecnologia. Dois

Diz-se que os computadores estão interconectados se forem capazes de trocar informações.

A conexão não precisa ser por meio de um fio de cobre; fibra ótica, microondas, infravermelho, e satélites de comunicação também podem ser usados. As redes vêm em vários tamanhos, formas e formas, como veremos mais tarde. Eles geralmente estão conectados juntos a fazer redes maiores, com a **Internet** sendo o exemplo mais conhecido de um rede de redes.

Há uma confusão considerável na literatura entre uma rede de computadores e um **sistema distribuído**. A principal diferença é que, em um sistema distribuído, um coleção de computadores independentes aparece para seus usuários como um único sistema coerente tem. Normalmente, ele possui um único modelo ou paradigma que apresenta aos usuários. Do-dez uma camada de software no topo do sistema operacional, chamada **middleware**, é responsável pela implementação deste modelo. Um exemplo bem conhecido de um sistema é a **World Wide Web**. Ele roda na Internet e apresenta um modelo no qual tudo parece um documento (página da Web).

Em uma rede de computadores, essa coerência, modelo e software estão ausentes. Comercial são expostos às máquinas reais, sem qualquer tentativa do sistema de fazer

as máquinas parecem e agem de maneira coerente. Se as máquinas têm diferentes software e diferentes sistemas operacionais, que são totalmente visíveis para os usuários. Se um usuário quer executar um programa em uma máquina remota, ele

tem que entrar nessa máquina e execute lá.

Com efeito, um sistema distribuído é um sistema de software construído em cima de uma rede.

O software confere um alto grau de coesão e transparência. Então, o distinção entre uma rede e um sistema distribuído reside no software (especialmente o sistema operacional), em vez do hardware.

No entanto, existe uma sobreposição considerável entre os dois assuntos. Para exemplo, tanto os sistemas distribuídos quanto as redes de computadores precisam mover arquivos por aí. A diferença está em quem invoca o movimento, o sistema ou o usuário.

† " Ele " deve ser lido como " ele ou ela " ao longo deste livro.

Página 27

SEC. 1,1
USOS DE REDES DE COMPUTADOR

3

Embora este livro se concentre principalmente em redes, muitos dos tópicos também são importantes importante em sistemas distribuídos. Para obter mais informações sobre sistemas distribuídos, ver Tanenbaum e Van Steen (2007).

1.1 USOS DE REDES DE COMPUTADORES

Antes de começarmos a examinar as questões técnicas em detalhes, vale a pena dedicar algum tempo para apontar por que as pessoas estão interessadas em redes de computadores e para que eles podem ser usados. Afinal, se ninguém estivesse interessado em redes de computadores obras, poucas delas seriam construídas. Começaremos com os usos tradicionais em empresas, em seguida, vá para a rede doméstica e desenvolvimentos recentes sobre usuários móveis e acabe com as questões sociais.

1.1.1 Aplicativos de negócios

A maioria das empresas possui um número significativo de computadores. Por exemplo, um a empresa pode ter um computador para cada trabalhador e usá-lo para projetar produtos, escrever brochuras e fazer a folha de pagamento. Inicialmente, alguns desses computadores podem ter

trabalhou isoladamente dos outros, mas em algum ponto, a gestão pode ter decidido conectá-los para poder distribuir informações em todo o empresa.

Colocado de uma forma um pouco mais geral, o problema aqui é o **compartilhamento de recursos**.

o objetivo é disponibilizar todos os programas, equipamentos e, principalmente, dados para qualquer pessoa na rede, independentemente da localização física do recurso ou do usuário.

Um exemplo óbvio e comum é ter um grupo de funcionários de escritório compartilhando um impressora comum. Nenhum dos indivíduos realmente precisa de uma impressora particular e um impressoras em rede de alto volume costumam ser mais baratas, rápidas e fáceis de manter do que uma grande coleção de impressoras individuais.

No entanto, provavelmente ainda mais importante do que compartilhar recursos físicos como como impressoras e sistemas de backup de fita, está compartilhando informações. Pequenas empresas

e grandes são vitalmente dependentes de informações computadorizadas. A maioria das empresas ter registros de clientes, informações de produtos, estoques, demonstrações financeiras, impostos informações e muito mais online. Se todos os seus computadores parassem de repente, um banco não poderia durar mais de cinco minutos. Uma moderna fábrica, com uma linha de montagem controlada por computador não duraria nem 5 segundos. Mesmo um pequeno

agência de viagens ou escritório de advocacia de três pessoas agora é altamente dependente da rede de computadores

funciona para permitir que os funcionários acessem informações e documentos relevantes imediatamente.

Para empresas menores, todos os computadores provavelmente estão em um único escritório ou talvez um único edifício, mas para os maiores, os computadores e funcionários podem estar espalhados por dezenas de escritórios e fábricas em muitos países. No entanto, um vendedor em Nova York às vezes pode precisar de acesso a um estoque de produtos

Página 28

4

INTRODUÇÃO

INDIVÍDUO. 1

banco de dados em Cingapura. Redes chamadas **VPNs** (**Redes Privadas Virtuais**) podem ser usado para juntar as redes individuais em locais diferentes em uma rede estendida trabalhos. Em outras palavras, o simples fato de um usuário estar a 15.000 km de distância de seus dados não deve impedi-lo de usar os dados como se fossem local. Este objetivo pode ser resumido dizendo que é uma tentativa de acabar com o " tirania da geografia. "

Em termos mais simples, pode-se imaginar o sistema de informação de uma empresa como consistindo em um ou mais bancos de dados com informações da empresa e algum número de funcionários que precisam acessá-los remotamente. Neste modelo, os dados são armazenados ed em computadores poderosos chamados **servidores** . Muitas vezes, estes são alojados centralmente e

mantido por um administrador de sistema. Em contraste, os funcionários têm máquinas, chamadas **clientes** , em suas mesas, com as quais acessam dados remotos, para exemplo, para incluir nas planilhas que estão construindo. (Às vezes vamos referir-se ao usuário humano da máquina cliente como o " cliente ", mas deveria ser claro do contexto, quer nos referimos ao computador ou ao seu usuário.) O cliente e as máquinas servidoras são conectadas por uma rede, conforme ilustrado na Figura 1-1. Observe que mostramos a rede como um oval simples, sem nenhum detalhe. Vamos usar isso forma quando nos referimos a uma rede no sentido mais abstrato. Quando mais detalhes são necessário, ele será fornecido.

Cliente
Servidor
Rede

Figura 1-1. Uma rede com dois clientes e um servidor.

Todo esse arranjo é chamado de **modelo cliente-servidor** . É amplamente utilizado e forma a base de grande parte do uso da rede. A percepção mais popular é que de um **aplicativo da Web** , em que o servidor gera páginas da Web com base em seus dados base em resposta às solicitações do cliente que podem atualizar o banco de dados. O cliente-servidor modelo é aplicável quando o cliente e o servidor estão no mesmo prédio (e pertencem à mesma empresa), mas também quando estão distantes. Por exemplo, quando uma pessoa em casa acessa uma página na World Wide Web, o mesmo modelo é empregado, com o servidor Web remoto sendo o servidor e o pessoal do usuário

Página 29

SEC. 1,1

USOS DE REDES DE COMPUTADOR

5

computador sendo o cliente. Na maioria das condições, um servidor pode lidar com um grande número (centenas ou milhares) de clientes simultaneamente.

Se olharmos para o modelo cliente-servidor em detalhes, vemos que dois processos (ou seja, programas em execução) estão envolvidos, um na máquina cliente e um no servidor máquina. A comunicação assume a forma do processo do cliente enviando uma mensagem pela rede para o processo do servidor. O processo do cliente então espera por uma resposta mensagem. Quando o processo do servidor obtém a solicitação, ele executa a solicitação trabalhar ou procurar os dados solicitados e enviar uma resposta. Essas mensagens são mostrado na Fig. 1-2.

Processo do cliente

Processo de servidor
Máquina cliente
Rede
Resposta
Solicitação
Máquina servidor

Figura 1-2. O modelo cliente-servidor envolve solicitações e respostas.

Um segundo objetivo de configurar uma rede de computadores tem a ver com as pessoas, em vez do que informação ou mesmo computadores. Uma rede de computadores pode fornecer um poderoso **meio de comunicação** entre os funcionários. Praticamente todas as empresas que têm dois ou mais computadores agora possuem **e - mail** (**correio eletrônico**), que os funcionários geram

almente use para uma grande comunicação diária. Na verdade, uma queixa comum em torno o refrigerador de água é a quantidade de e-mails com os quais todos precisam lidar, muitos deles bastante

sem sentido porque os chefes descobriram que podem enviar o mesmo (muitas vezes sem conteúdo) para todos os seus subordinados com o premir de um botão.

Chamadas telefônicas entre funcionários podem ser realizadas pela rede de computadores

em vez de pela companhia telefônica. Esta tecnologia é chamada de **telefonia IP** ou

Voz sobre IP (VoIP) quando a tecnologia da Internet é usada. O microfone e alto-falante em cada extremidade pode pertencer a um telefone habilitado para VoIP ou a comunicação do funcionário

puter. As empresas consideram essa uma ótima maneira de economizar nas contas de telefone.

Outras formas mais ricas de comunicação são possibilitadas pela rede de computadores trabalho. O vídeo pode ser adicionado ao áudio para que os funcionários em locais distantes possam ver

e ouvir uns aos outros durante uma reunião. Esta técnica é uma ferramenta poderosa para eliminando o custo e o tempo anteriormente dedicado à viagem. **O compartilhamento da área de trabalho** permite

trabalhadores remotos veem e interagem com uma tela gráfica de computador. Isso torna fácil para duas ou mais pessoas que trabalham distantes ler e escrever um texto negro compartilhado bordo ou escrever um relatório juntos. Quando um trabalhador muda para um online documento, os outros podem ver a mudança imediatamente, em vez de esperar vários dias para uma carta. Tal aceleração torna a cooperação entre grupos distantes de pessoas fáceis onde antes era impossível. Formas mais ambiciosas de coordenação remota, como telemedicina, só agora estão começando a ser usados (por exemplo,

Página 30

6

INTRODUÇÃO

INDIVÍDUO. 1

monitoramento remoto do paciente), mas pode se tornar muito mais importante. É algumas vezes disse que comunicação e transporte estão tendo uma corrida, e que- qualquer vitória tornará o outro obsoleto.

Um terceiro objetivo para muitas empresas é fazer negócios eletronicamente, especialmente com clientes e fornecedores. Este novo modelo é denominado **e-commerce** (**eletrônico comércio**) e tem crescido rapidamente nos últimos anos. Companhias aéreas, livrarias e outros varejistas descobriram que muitos clientes gostam da conveniência da loja ping de casa. Consequentemente, muitas empresas fornecem catálogos de seus produtos e serviços online e receba pedidos online. Fabricantes de automóveis, artesanato e computadores, entre outros, compram subsistemas de uma variedade de fornecedores e depois monte as peças. Usando redes de computadores, os fabricantes podem colocar pedidos eletronicamente conforme necessário. Isso reduz a necessidade de grandes estoques e aumenta a eficiência.

1.1.2 Aplicativos domésticos

Em 1977, Ken Olsen foi presidente da Digital Equipment Corporation, então o segundo fornecedor de computadores do mundo (depois da IBM). Quando perguntado por que Dig-

A ital não estava indo muito atrás do mercado de computadores pessoais, ele disse:

" Não há razão para ninguém ter um computador em casa. " História mostrou o contrário e Digital não existe mais. As pessoas inicialmente compraram computadores para processamento de texto e jogos. Recentemente, o maior motivo para comprar uma casa o computador provavelmente era para acesso à Internet. Agora, muitos dispositivos eletrônicos de consumo,

como decodificadores, consoles de jogos e rádios-relógios, vêm com computadores e redes de computadores, especialmente redes sem fio e redes domésticas as obras são amplamente utilizadas para entretenimento, incluindo ouvir, olhar e criação de música, fotos e vídeos.

O acesso à Internet fornece aos usuários domésticos **conectividade** com computadores remotos. Tal como acontece com as empresas, os usuários domésticos podem acessar informações, comunicar-se com outros

pessoas e comprar produtos e serviços com e-commerce. O principal benefício agora vem da conexão fora de casa. Bob Metcalfe, o inventor do Ethernet, a hipótese de que o valor de uma rede é proporcional ao quadrado do número de usuários porque este é aproximadamente o número de conexões diferentes que pode ser feito (Gilder, 1993). Esta hipótese é conhecida como " lei de Metcalfe ". ajuda a explicar como a enorme popularidade da Internet vem de seu Tamanho.

O acesso a informações remotas vem de várias formas. Pode ser surfar no World Wide Web para obter informações ou apenas para se divertir. As informações disponíveis incluem as artes, negócios, culinária, governo, saúde, história, hobbies, recreação, ciências cia, esportes, viagens e muitos outros. Diversão vem de muitas maneiras para mencionar, além de algumas maneiras que é melhor não mencionar.

Muitos jornais estão online e podem ser personalizados. Por exemplo, às vezes é possível dizer a um jornal que você quer tudo sobre corrupção

Página 31

SEC. 1,1

USOS DE REDES DE COMPUTADOR

7

políticos, grandes incêndios, escândalos envolvendo celebridades e epidemias, mas sem pé-bola, obrigado. Às vezes é possível fazer o download dos artigos selecionados para o seu computador enquanto você dorme. Conforme esta tendência continua, isso causará desemprego entre os jornaleiros de 12 anos, mas os jornais gostam porque des- tribuição sempre foi o elo mais fraco de toda a cadeia produtiva. Do claro, para fazer este modelo funcionar, eles primeiro terão que descobrir como fazer dinheiro neste novo mundo, algo não totalmente óbvio, uma vez que os usuários da Internet espere que tudo seja gratuito.

O próximo passo além dos jornais (além de revistas e periódicos científicos) é a biblioteca digital online. Muitas organizações profissionais, como a ACM (www.acm.org) e a IEEE Computer Society (www.computer.org), já ter todos os seus jornais e anais de conferências online. Livro eletrônico lido ers e bibliotecas online podem tornar os livros impressos obsoletos. Os célicos devem tomar observe o efeito que a imprensa teve sobre o manuscrito iluminado medieval.

Muitas dessas informações são acessadas usando o modelo cliente-servidor, mas lá é um modelo diferente e popular para acessar informações que atendem pelo nome de **comunicação ponto a ponto** (Parameswaran et al., 2001). Nesta forma, indivíduos Todos os que formam um grupo livre podem se comunicar com outras pessoas do grupo, como mostrado

na Fig. 1-3. Cada pessoa pode, em princípio, comunicar-se com um ou mais outros pessoas; não há divisão fixa em clientes e servidores.

Figura 1-3. Em um sistema ponto a ponto, não há clientes e servidores fixos.

Muitos sistemas ponto a ponto, como o BitTorrent (Cohen, 2003), não têm nenhum banco de dados central de conteúdo. Em vez disso, cada usuário mantém seu próprio banco de dados localmente

e fornece uma lista de outras pessoas próximas que são membros do sistema. Um novo o usuário pode ir a qualquer membro existente para ver o que ele tem e obter os nomes de outros membros para inspecionar por mais conteúdo e mais nomes. Este processo de pesquisa pode ser repetido indefinidamente para construir um grande banco de dados local do que está lá fora. É uma atividade que se tornaria tediosa para as pessoas, mas os computadores são excelentes nisso.

Página 32

8

INTRODUÇÃO INDIVÍDUO. 1

A comunicação ponto a ponto costuma ser usada para compartilhar músicas e vídeos. É realmente atingiu o grande momento por volta de 2000 com um serviço de compartilhamento de música chamado Napster que foi encerrado após o que foi provavelmente o maior caso de violação de direitos autorais em todos da história registrada (Lam e Tan, 2001; e Macedônia, 2000). Aplicação legal também existem opções para comunicação ponto a ponto. Isso inclui fãs que compartilham pub-música de domínio lic, famílias compartilhando fotos e filmes e usuários fazendo download pacotes de software público. Na verdade, um dos aplicativos de Internet mais populares ofall, email, é inherentemente peer-to-peer. Esta forma de comunicação provavelmente crescer consideravelmente no futuro.

Todos os aplicativos acima envolvem interações entre uma pessoa e um re-banco de dados mote cheio de informações. A segunda categoria ampla de uso de rede é comunicação pessoa a pessoa, basicamente a resposta do século 21 ao 19 telefone do século. O e-mail já é usado diariamente por milhões de pessoas em todo o mundo e seu uso está crescendo rapidamente. Ele já contém rotineiramente áudio e vídeo, bem como texto e imagens. O cheiro pode demorar um pouco.

Qualquer adolescente que se preze é viciado em **mensagens instantâneas**. Isto facilidade, derivada do programa UNIX *talk* em uso desde cerca de 1970, permite dois pessoas digitem mensagens umas para as outras em tempo real. Existem mensagens com várias pessoas

serviços de saging também, como o serviço **Twitter** que permite que as pessoas enviem textos curtos mensagens chamadas " tweets " para seu círculo de amigos ou outro público interessado.

A Internet pode ser usada por aplicativos para transportar áudio (por exemplo, rádio da Internet estações) e vídeo (por exemplo, YouTube). Além de ser uma forma barata de ligar para pessoas distantes

amigos, esses aplicativos podem fornecer experiências ricas, como telelearning, o que significa que irão 8 Uma . M . aulas sem o inconveniente de ter que sair da cama primeiro. No longo prazo, o uso de redes para aprimorar de pessoa para pessoa a comunicação pode ser mais importante do que qualquer uma das outras. Pode se tornar extremamente importante para as pessoas com desafios geográficos, dando-lhes o mesmo acesso aos serviços que as pessoas que vivem no meio de uma grande cidade.

Entre as comunicações pessoa a pessoa e as informações de acesso estão aplicativos de **rede social**. Aqui, o fluxo de informações é impulsionado pela relação relações que as pessoas declaram entre si. Uma das redes sociais mais populares sites de relacionamento é o **Facebook**. Permite que as pessoas atualizem seus perfis pessoais e compartilha as atualizações com outras pessoas que eles declararam serem seus amigos.

Outros aplicativos de rede social podem fazer apresentações por meio de amigos de amigos, envie mensagens de notícias para amigos como o Twitter acima e muito mais.

Ainda mais vagamente, grupos de pessoas podem trabalhar juntos para criar conteúdo. UMA **wiki**, por exemplo, é um site colaborativo que os membros de uma comunidade editar. O wiki mais famoso é a **Wikipedia**, uma enciclopédia que qualquer um pode editar, mas existem milhares de outros wikis.

Nossa terceira categoria é o comércio eletrônico no sentido mais amplo do termo.

Compras em casa já é popular e permite que os usuários inspecionem os catálogos online de milhares de empresas. Alguns desses catálogos são interativos, mostrando dutos em diferentes pontos de vista e em configurações que podem ser personalizadas.

SEC. 1,1
USOS DE REDES DE COMPUTADOR

9

Depois que o cliente compra um produto eletronicamente, mas não consegue descobrir como usá-lo nele, o suporte técnico online pode ser consultado.

Outra área em que o e-commerce é amplamente utilizado é o acesso a instituições financeiras tutions. Muitas pessoas já pagam suas contas, gerenciam suas contas bancárias e lidar com seus investimentos eletronicamente. Esta tendência certamente continuará como as obras ficam mais seguras.

Uma área que virtualmente ninguém previu são os mercados de pulgas eletrônicos (e-pulgas?). Os leilões online de produtos em segunda mão tornaram-se uma grande indústria. Ao contrário e-commerce tradicional, que segue o modelo cliente-servidor, leilões online são ponto a ponto no sentido de que os consumidores podem atuar como compradores e vendedores. Algumas dessas formas de comércio eletrônico adquiriram pequenas tags atraentes com base em o fato de que " para " e " 2 " são pronunciados da mesma forma. Os mais populares são listado na Fig. 1-4.

Tag

Nome completo

Exemplo

B2C

Business-to-consumer

Encomendar livros online

B2B

De empresa para empresa

Fabricante de automóveis encomenda pneus do fornecedor

G2C

Governo para consumidor

Governo distribuindo formulários fiscais eletronicamente

C2C

Consumidor para consumidor

Leilão de produtos usados online

P2P

Pessoa para pessoa

Compartilhamento de música

Figura 1-4. Algumas formas de comércio eletrônico.

Nossa quarta categoria é o entretenimento. Isso fez grandes avanços em casa nos últimos anos, com a distribuição de programas de música, rádio e televisão, e os filmes na Internet começam a rivalizar com os dos mecanismos tradicionais. Comercial pode encontrar, comprar e baixar músicas em MP3 e filmes com qualidade de DVD e adicioná-los para sua coleção pessoal. Os programas de TV agora chegam a muitos lares via **IPTV (IP TeleVision)** sistemas baseados em tecnologia IP em vez de TV a cabo ou rádio transmissões. Os aplicativos de streaming de mídia permitem que os usuários sintonizem estações de rádio da Internet

ções ou assistir episódios recentes de seus programas de TV favoritos. Naturalmente, tudo isso o conteúdo pode ser movido pela sua casa entre diferentes dispositivos, monitores e alto-falantes, geralmente com uma rede sem fio.

Em breve, será possível pesquisar qualquer filme ou programa de televisão feitos, em qualquer país, e exibidos na tela instantaneamente. Novos filmes pode se tornar interativo, onde o usuário é ocasionalmente solicitado a contar a história direção (deve Macbeth assassinar Duncan ou apenas esperar a hora certa?) com alternativa cenários fornecidos para todos os casos. A televisão ao vivo também pode se tornar interativa, com o público participando de programas de perguntas e respostas, escolhendo entre os concorrentes, e assim em.

Outra forma de entretenimento é o jogo. Já temos várias pessoas jogos de simulação em tempo real, como esconde-esconde em uma masmorra virtual e vôo

INTRODUÇÃO

INDIVÍDUO. 1

simuladores com os jogadores de uma equipe tentando derrubar os jogadores do equipe adversária. Os mundos virtuais fornecem um ambiente persistente em que milhares de os usuários podem experimentar uma realidade compartilhada com gráficos tridimensionais. Nossa última categoria é a **computação ubíqua**, na qual a computação está incorporada na vida cotidiana, como na visão de Mark Weiser (1991). Muitas casas são todas pronto com sistemas de segurança que incluem sensores de porta e janela, e existem muitos mais sensores que podem ser dobrados em um monitor doméstico inteligente, como consumo de energia. Seus medidores de eletricidade, gás e água também podem relatar uso na rede. Isso economizaria dinheiro, pois não haveria necessidade de enviar leitores de medidores. E seus detectores de fumaça podem chamar o corpo de bombeiros em vez de fazer muito barulho (que tem pouco valor se não houver ninguém em casa). Enquanto o o custo de detecção e comunicação cai, mais e mais medição e re-a portabilidade será feita com redes.

Cada vez mais, dispositivos eletrônicos de consumo estão em rede. Por exemplo, alguns câmeras de última geração já possuem um recurso de rede sem fio e o utilizam para enviar fotos em um monitor próximo para visualização. Fotógrafos profissionais de esportes podem também enviam suas fotos para seus editores em tempo real, primeiro sem fio para um acesso aponte então pela Internet. Dispositivos como televisores que se conectam à parede pode usar **redes de linha de energia** para enviar informações por toda a casa pelo fios que transportam eletricidade. Pode não ser muito surpreendente ter esses objetos a rede, mas os objetos que não pensamos como computadores podem sentir e compreender comunicar informações também. Por exemplo, seu chuveiro pode registrar o uso de água, dar feedback visual enquanto você se ensaboar e relatar para o ambiente doméstico aplicativo de monitoramento quando você estiver pronto para ajudar a economizar na conta de água. Uma tecnologia chamada **RFID (Radio Frequency IDentification)** vai empurrar isso ideia ainda mais no futuro. As etiquetas RFID são chips passivos (ou seja, não têm bateria) o tamanho dos selos e já podem ser afixados em livros, passaportes, animais de estimação, crédito cartões e outros itens em casa e fora. Isso permite que os leitores RFID localizem e comunicar-se com os itens a uma distância de até vários metros, dependendo o tipo de RFID. Originalmente, o RFID foi comercializado para substituir os códigos de barras. isto ainda não teve sucesso porque os códigos de barras são gratuitos e as etiquetas RFID custam alguns centavos.

Claro, as etiquetas RFID oferecem muito mais e seu preço está diminuindo rapidamente. Eles pode transformar o mundo real na Internet das coisas (ITU, 2005).

1.1.3 Usuários móveis

Os computadores móveis, como laptops e computadores de mão, são um dos segmentos de crescimento mais rápido da indústria de computadores. Suas vendas já ultrapassou os de computadores desktop. Por que alguém iria querer um? Pessoas em muitas vezes querem usar seus dispositivos móveis para ler e enviar e-mail, tweetar, assistir filmes, baixar músicas, jogar ou simplesmente navegar na Web para obter informações. Eles querem fazer todas as coisas que fazem em casa e no escritório. Naturalmente, eles querem fazê-los de qualquer lugar na terra, no mar ou no ar.

A **conectividade** com a Internet permite muitos desses usos móveis. Desde ter uma conexão com fio é impossível em carros, barcos e aviões, há muitos interesse em redes sem fio. Redes celulares operadas por telefone com empresas são um tipo familiar de rede sem fio que nos cobre com cobertura para telefones celulares. Pontos de **acesso sem fio** baseados no padrão 802.11 são outro tipo de rede sem fio para computadores móveis. Eles surgiram em todos os lugares que as pessoas vão, resultando em uma colcha de retalhos de cobertura em cafés, hotéis, aeroportos, escolas, trens e aviões. Qualquer pessoa com um laptop e um modem sem fio

pode simplesmente ligar o computador e estar conectado à Internet através do hotspot, como se o computador estivesse conectado a uma rede com fio.

As redes sem fio são de grande valor para frotas de caminhões, táxis, veículos de entrega cles e reparadores para manter contato com sua base doméstica. Por exemplo, em muitas cidades, os motoristas de táxi são empresários independentes, ao invés de serem empregados de uma empresa de táxis. Em algumas dessas cidades, os táxis têm um display de o motorista pode ver. Quando um cliente liga, um despachante central digita na coleta e pontos de destino. Esta informação é exibida nos visores dos motoristas e um bipe soa. O primeiro motorista a apertar um botão na tela recebe a chamada.

As redes sem fio também são importantes para os militares. Se você tem que ser capaz de lutar uma guerra em qualquer lugar da Terra em curto prazo, contando com o uso da rede local infraestrutura funcional provavelmente não é uma boa ideia. É melhor trazer o seu próprio.

Embora a rede sem fio e a computação móvel estejam frequentemente relacionadas, eles não são idênticos, como mostra a Figura 1-5. Aqui vemos uma distinção entre **fixa** redes **sem fio** e **móveis sem fio**. Mesmo os notebooks às vezes são com fio. Por exemplo, se um viajante conectar um notebook à rede com fio conector de trabalho em um quarto de hotel, ele tem mobilidade sem uma rede sem fio.

Sem fio

Móvel

Aplicações típicas

Não

Não

Computadores desktop em escritórios

Não

sim

Um notebook usado em um quarto de hotel

sim

Não

Redes em edifícios sem fio

sim

sim

Armazene o estoque com um computador portátil

Figura 1-5. Combinacões de redes sem fio e computação móvel.

Por outro lado, alguns computadores sem fio não são móveis. Em casa e em escritórios ou hotéis que não têm cabeamento adequado, pode ser mais conveniente conectar computadores desktop ou media players sem fio do que para instalar fios. Instalando um rede sem fio pode exigir pouco mais do que comprar uma pequena caixa com alguns tronics nele, desempacotando-o e conectando-o. Esta solução pode ser muito mais barata do que ter operários instalando dutos de cabos para fazer a fiação do prédio. Finalmente, também existem verdadeiros aplicativos móveis e sem fio, como pessoas que caminham circulando pelas lojas com computadores portáteis registrando o inventário. Em muitos ocupado

12

INTRODUÇÃO

INDIVÍDUO. 1

aeroportos, funcionários de devolução de aluguel de automóveis trabalham no estacionamento com comunicação móvel sem fio

puters. Eles escaneiam os códigos de barras ou chips RFID de carros devolvidos e seus celulares dispositivo, que possui uma impressora embutida, liga para o computador principal, obtém as informações de aluguel

mação e imprime a fatura no local.

Talvez o principal motivador dos aplicativos móveis sem fio seja o telefone móvel.

Mensagens de texto ou **mensagens de texto** são extremamente populares. Permite que um usuário de telefone celular

digite uma mensagem curta que é então entregue pela rede celular a outro assinante móvel. Poucas pessoas teriam previsto dez anos atrás que ter adolescentes digitando tediosamente mensagens de texto curtas em telefones celulares seria um imenso fabricante de dinheiro para companhias telefônicas. Mas mensagens de texto (ou **mensagem curta**

O serviço, como é conhecido fora dos EUA) é muito lucrativo, pois custa à transportadora mas uma pequena fração de um centavo para transmitir uma mensagem de texto, um serviço para o qual cobrar muito mais.

A tão esperada convergência dos telefones e da Internet finalmente chegou, e vai acelerar o crescimento dos aplicativos móveis. **Telefones inteligentes**, como o popular iPhone, combinam aspectos de telefones celulares e dispositivos móveis puters. As redes celulares (3G e 4G) às quais eles se conectam podem fornecer serviços de dados para uso da Internet, bem como tratamento de chamadas telefônicas. Muitos anúncios telefones avançados se conectam a pontos de acesso sem fio também e alternam automaticamente entre redes para escolher a melhor opção para o usuário.

Outros dispositivos eletrônicos de consumo também podem usar redes celulares e de pontos de acesso para permanecer conectado a computadores remotos. Os leitores de livros eletrônicos podem baixar um livro recém-comprado ou a próxima edição de uma revista ou jornal de hoje onde quer que eles vaguem. Os porta-retratos eletrônicos podem atualizar suas exibições na hora com novas imagens.

Já que os telefones celulares sabem sua localização, muitas vezes porque estão equipados com receptores **GPS** (**Sistema de Posicionamento Global**), alguns serviços são intencionalmente dependente da localização. Mapas e direções para celular são candidatos óbvios como seu Telefone e carro com GPS provavelmente têm uma ideia melhor de onde você está do que você Faz. O mesmo acontece com as pesquisas por uma livraria próxima ou restaurante chinês, ou um local

visão do tempo. Outros serviços podem registrar a localização, como anotar fotos e vídeos com o local em que foram feitos. Esta anotação é conhecida como "geo-tagging."

Uma área em que os telefones celulares agora estão começando a ser usados é o **m-commerce** (**mobile-commerce**) (Senn, 2000). Mensagens curtas de texto do celular são usadas para autorizar o pagamento de alimentos em máquinas de venda automática, ingressos de cinema e outros

pequenos itens em vez de dinheiro e cartões de crédito. A cobrança, então, aparece no conta de telefone móvel. Quando equipado com **NFC** (**Near Field Communication**) tecnologia que o celular pode atuar como um smartcard RFID e interagir com um próximo leitor para pagamento. As forças motrizes por trás desse fenômeno são os dispositivos móveis fabricantes de dispositivos e operadores de rede, que estão se esforçando para descobrir como obter um pedaço do bolo do comércio eletrônico. Do ponto de vista da loja, este esquema pode economizar a maior parte da taxa da empresa de cartão de crédito, que pode chegar a vários por cento.

Página 37

SEC. 1,1
USOS DE REDES DE COMPUTADOR

13

Claro, este plano pode sair pela culatra, uma vez que os clientes em uma loja podem usar o RFID ou leitores de código de barras em seus dispositivos móveis para verificar os preços dos concorrentes antes comprá-los e usá-los para obter um relatório detalhado sobre onde mais um item pode ser comprado perseguido nas proximidades e a que preço.

Uma grande vantagem do m-commerce é que os usuários de telefones celulares estão acostumados a pagar por tudo (ao contrário dos usuários da Internet, que esperam tudo para ser gratuito). Se um site da Internet cobrasse uma taxa para permitir que seus clientes para pagar com cartão de crédito, haveria um barulho enorme de uivos dos usuários.

Se, no entanto, uma operadora de telefonia móvel seus clientes pagarem por itens em uma loja, acenar com o telefone na caixa registradora e, em seguida, adicionar uma taxa por esta conveniência

uma vez, provavelmente seria aceito como normal. O tempo vai dizer.

Sem dúvida, o uso de computadores móveis e sem fio crescerá rapidamente no futuro conforme o tamanho dos computadores diminui, provavelmente de maneiras que ninguém pode prever agora.

Vamos dar uma olhada rápida em algumas possibilidades. **Redes de sensores** são compostas de nós que coletam e transmitem sem fio as informações que percebem sobre o estado do mundo físico. Os nós podem fazer parte de itens familiares, como carros ou telefones, ou podem ser pequenos dispositivos separados. Por exemplo, seu carro pode coletar dados em sua localização, velocidade, vibração e eficiência de combustível de seu diagnóstico a bordo sistema e enviar essas informações para um banco de dados (Hull et al., 2006). Esses dados pode ajudar a encontrar buracos, planejar viagens em estradas congestionadas e dizer se você é um "bebedor de gasolina" em comparação com outros motoristas no mesmo trecho da estrada.

As redes de sensores estão revolucionando a ciência, fornecendo uma grande quantidade de dados sobre

comportamento que não poderia ser observado anteriormente. Um exemplo é monitorar o migração de zebras individuais, colocando um pequeno sensor em cada animal (Juang et al., 2002). Os pesquisadores embalaram um computador sem fio em um cubo de 1 mm borda (Warneke et al., 2001). Com computadores móveis tão pequenos, até pequenos pássaros, roedores e insetos podem ser rastreados.

Mesmo os usos mundanos, como parquímetros, podem ser significativos porque eles fazem uso de dados que não estavam disponíveis anteriormente. Parquímetros sem fio pode aceitar pagamentos com cartão de crédito ou débito com verificação instantânea via wireless ligação. Eles também podem relatar quando estão em uso na rede sem fio. Isto permitiria que os motoristas baixassem um mapa de estacionamento recente para seus carros para que pudessem encontrar um

local disponível mais facilmente. Claro, quando um medidor expira, ele também pode verificar para a presença de um carro (refletindo um sinal dele) e relatar a expiração para fiscalização do estacionamento. Estima-se que as prefeituras dos Estados Unidos sozinho poderia arrecadar US \$ 10 bilhões adicionais desta forma (Harte et al., 2000).

Os computadores vestíveis são outra aplicação promissora. Relógios inteligentes com rádios fazem parte de nosso espaço mental desde sua aparição no Dick História em quadrinhos de Tracy em 1946; agora você pode comprá-los. Outros dispositivos podem ser

implantados, como marca-passos e bombas de insulina. Alguns deles podem ser contra rastreado em uma rede sem fio. Isso permite que os médicos os testem e reconfigurem mais facilmente. Também pode levar a alguns problemas desagradáveis se os dispositivos forem tão inseguros quanto

o PC médio e pode ser facilmente hackeado (Halperin et al., 2008).

14

INTRODUÇÃO INDIVÍDUO. 1

1.1.4 Questões Sociais

Redes de computadores, como a impressora de 500 anos atrás, permitem cidadãos distribuam e visualizem o conteúdo de maneiras que não eram possíveis anteriormente. Mas junto com o bom vem o ruim, pois esta liberdade recém-descoberta traz consigo muitas questões sociais, políticas e éticas não resolvidas. Vamos apenas mencionar brevemente um alguns deles; um estudo completo exigiria um livro completo, pelo menos.

Redes sociais, quadros de mensagens, sites de compartilhamento de conteúdo e uma série de outros aplicativos

aplicações permitem que as pessoas compartilhem suas opiniões com pessoas que pensam da mesma forma. Tanto tempo

já que os assuntos são restritos a tópicos técnicos ou hobbies, como jardinagem, não muito muitos problemas surgirão.

O problema vem com tópicos que realmente interessam às pessoas, como política,

religião ou sexo. As visualizações postadas publicamente podem ser profundamente ofensivas para alguns

pessoas. Pior ainda, eles podem não ser politicamente corretos. Além disso, opiniões não precisa se limitar ao texto; fotografias coloridas de alta resolução e videoclipes são facilmente compartilhado em redes de computadores. Algumas pessoas têm uma visão viva e deixe viver,

mas outros acham que postar certo material (por exemplo, ataques verbais a determinados países ou religiões, pornografia, etc.) é simplesmente inaceitável e que tal o conteúdo deve ser censurado. Diferentes países têm leis diferentes e conflitantes nesta área. Assim, o debate se intensifica.

No passado, as pessoas processaram as operadoras de rede, alegando que elas estão patrocinados pelo conteúdo do que carregam, assim como jornais e revistas estão. A resposta inevitável é que uma rede é como uma companhia telefônica ou os correios e não se pode esperar que policie o que seus usuários dizem.

Agora deve ser uma pequena surpresa saber que algumas redes operam todos bloqueiam conteúdo por seus próprios motivos. Alguns usuários de aplicativos ponto a ponto teve seu serviço de rede interrompido porque as operadoras de rede não acharam pro aptos a transportar grandes quantidades de tráfego enviadas por esses aplicativos. Essa os mesmos operadores provavelmente gostariam de tratar empresas diferentes de maneira diferente. E se

você é uma grande empresa e paga bem, então você recebe um bom serviço, mas se você é um jogador pequeno, você recebe um serviço ruim. Os oponentes desta prática argumentam que ponto a ponto e outros conteúdos devem ser tratados da mesma forma porque são todos apenas bits para a rede. Este argumento para comunicações que não são diferentes diferenciados por seu conteúdo ou fonte ou quem está fornecendo o conteúdo é conhecido como **neutralidade da rede** (Wu, 2003). É provavelmente seguro dizer que este debate irá por um tempo.

Muitas outras partes estão envolvidas na disputa pelo conteúdo. Por exemplo, pi-músicas e filmes classificados alimentaram o crescimento massivo de redes ponto a ponto, o que não agradou aos detentores dos direitos autorais, que ameaçaram (e às vezes tomadas) ação legal. Agora existem sistemas automatizados que pesquisam ponto a ponto redes e disparar avisos para operadores de rede e usuários suspeitos de violação de direitos autorais. Nos Estados Unidos, esses avisos são conhecidos como **Avisos de remoção do DMCA** após o **Digital Millennium Copyright Act**. Isto

Página 39

SEC. 1,1

USOS DE REDES DE COMPUTADOR

15

a busca é uma corrida armamentista porque é difícil detectar com segurança a violação de direitos autorais.

Até a sua impressora pode ser confundida com a culpada (Piatek et al., 2008).

As redes de computadores facilitam a comunicação. Eles também fazem isso fácil para as pessoas que administram a rede espionarem o tráfego. Isso configura discorda sobre questões como direitos do empregado versus direitos do empregador. Muitas pessoas ler e escrever e-mails no trabalho. Muitos empregadores reivindicaram o direito de ler e possivelmente censurar mensagens de funcionários, incluindo mensagens enviadas de um computador doméstico

computador fora do horário de trabalho. Nem todos os funcionários concordam com isso, especialmente o lat-ter parte.

Outro conflito está centrado em torno do governo contra os direitos dos cidadãos. O FBI instalou sistemas em muitos provedores de serviços de Internet para espionar todos os e-mail de entrada e saída para pepitas de interesse. Um sistema inicial foi original finalmente chamado de Carnivore, mas a má publicidade fez com que fosse renomeado para mais aparentemente inocente DCS1000 (Blaze e Bellovin, 2000; Sobel, 2001; e Zacks, 2001). O objetivo de tais sistemas é espionar milhões de pessoas na esperança de

talvez encontrando informações sobre atividades ilegais. Infelizmente para os espiões, a Quarta Emenda da Constituição dos EUA proíbe buscas do governo sem um mandado de busca, mas o governo frequentemente o ignora. Claro, o governo não tem o monopólio de ameaçar as pessoas privacidade. O setor privado também faz sua parte ao **traçar o perfil dos usuários**. Por exemplo, pequenos arquivos chamados **cookies** que os navegadores da Web armazenam nos computadores dos usuários permitem empresas para rastrear as atividades dos usuários no ciberespaço e também podem permitir cartão de crédito números, números de previdência social e outras informações confidenciais para vazar todos pela Internet (Berghel, 2001). Empresas que fornecem serviços baseados na Web podem manter grandes quantidades de informações pessoais sobre seus usuários que permitem para estudar as atividades do usuário diretamente. Por exemplo, o Google pode ler seu e-mail e mostrar anúncios com base em seus interesses, se você usar o serviço de e-mail, **Gmail**.

Uma nova reviravolta com dispositivos móveis é a privacidade de localização (Beresford e Stajano, 2003). Como parte do processo de prestação de serviço ao seu dispositivo móvel, a rede os operadores de trabalho aprendem onde você está em horários diferentes do dia. Isso permite que eles rastreie seus movimentos. Eles podem saber qual boate você frequenta e quais centro médico que você visita.

As redes de computadores também oferecem o potencial de aumentar a privacidade ao enviar mensagens anônimas. Em algumas situações, esse recurso pode ser desejável. Além de impedir que as empresas aprendam seus hábitos, oferece, por exemplo, uma forma de estudantes, soldados, funcionários e cidadãos denunciarem comportamento por parte de professores, oficiais, superiores e políticos sem medo de represálias. Por outro lado, nos Estados Unidos e na maioria das outras democracias, a lei permite especificamente a uma pessoa acusada o direito de confrontar e desafiar seu acusador no tribunal, então acusações anônimas não podem ser usadas como prova. A Internet torna possível encontrar informações rapidamente, mas muitas é mal considerado, enganoso ou totalmente errado. Esse conselho médico você

Página 40

16

INTRODUÇÃO

INDIVÍDUO. 1

extraído da Internet sobre a dor no peito pode ter vindo de um Vencedor do Prêmio Nobel ou de um aluno que abandonou o ensino médio. Outras informações são freqüentemente indesejadas. O lixo eletrônico (spam) tem tornou-se parte da vida porque os spammers coletaram milhões de endereços de e-mail - es e aspirantes a profissionais de marketing podem enviar mensagens geradas por computador para eles. A inundação de spam resultante rivaliza com o fluxo de mensagens de pessoas reais. Felizmente, o software de filtragem é capaz de ler e descartar o spam gerado por outros computadores, com maior ou menor grau de sucesso. Ainda outro conteúdo se destina a comportamento criminoso. Páginas da web e e-mail mensagens contendo conteúdo ativo (basicamente, programas ou macros que executam em máquina do receptor) pode conter vírus que invadem o seu computador. Eles pode ser usado para roubar as senhas da sua conta bancária ou para que o seu computador enviar spam como parte de um **botnet** ou pool de máquinas comprometidas. As mensagens de **phishing** disfarçam-se como originárias de uma parte confiável, por exemplo, seu banco, para tentar induzi-lo a revelar informações confidenciais, para por exemplo, números de cartão de crédito. O roubo de identidade está se tornando um problema sério, pois ladrões coletam informações suficientes sobre a vítima para obter cartões de crédito e outros documentos em nome da vítima. Pode ser difícil evitar que os computadores se façam passar por pessoas no Internet. Este problema levou ao desenvolvimento de **CAPTCHAs**, em que um

computador pede a uma pessoa para resolver uma pequena tarefa de reconhecimento, por exemplo, digitando no

letras mostradas em uma imagem distorcida, para mostrar que são humanas (von Ahn, 2001).

Este processo é uma variação do famoso teste de Turing em que uma pessoa faz perguntas em uma rede para julgar se a entidade que está respondendo é humana.

Muitos desses problemas poderiam ser resolvidos se a indústria de computadores assumisse segurança do computador a sério. Se todas as mensagens foram criptografadas e autenticadas, seria mais difícil cometer travessuras. Essa tecnologia está bem estabelecida e nós vai estudá-lo em detalhes no cap. 8. O problema é que o hardware e o software os dois sabem que colocar recursos de segurança custa dinheiro e seus clientes não exigindo tais recursos. Além disso, um número substancial de problemas são causados por software com erros, o que ocorre porque os fornecedores continuam adicionando mais

e mais recursos para seus programas, o que inevitavelmente significa mais código e, portanto, mais bugs. Um imposto sobre novos recursos pode ajudar, mas pode ser difícil de vender alguns quartos. Um reembolso por software defeituoso pode ser bom, mas seria levou à falência toda a indústria de software no primeiro ano.

Redes de computadores levantam novos problemas jurídicos quando interagem com antigas leis. O jogo eletrônico é um exemplo. Os computadores têm simulado coisas por décadas, então por que não simular caça-níqueis, rodas de roleta, blackjack revendedores e mais equipamentos de jogo? Bem, porque é ilegal em muitos locais. O problema é que o jogo é legal em muitos outros lugares (Inglaterra, por exemplo) e os proprietários de cassinos perceberam o potencial para jogos de azar na Internet. O que acontece se o jogador, o cassino e o servidor estiverem todos em países diferentes tenta, com leis conflitantes? Boa pergunta.

Página 41

SEC. 1,2

HARDWARE DE REDE

17

1.2 HARDWARE DE REDE

Agora é hora de desviar nossa atenção dos aplicativos e aspectos sociais da rede (a sobremesa) para as questões técnicas envolvidas no projeto de rede (o espinafre). Não existe uma taxonomia geralmente aceita em que todas as redes de computador funcionam bem, mas duas dimensões se destacam como importantes: tecnologia de transmissão e escala. Vamos agora examinar cada um deles.

Em termos gerais, existem dois tipos de tecnologia de transmissão que estão em uso generalizado: links de **transmissão** e links **ponto a ponto**.

Links ponto a ponto conectam pares individuais de máquinas. Para ir de origem ao destino em uma rede composta de links ponto a ponto, mensagens curtas sábios, chamados de **pacotes** em certos contextos, podem ter que visitar primeiro um ou mais mediar máquinas. Muitas vezes, várias rotas, de comprimentos diferentes, são possíveis, então encontrar bons é importante em redes ponto a ponto. Ponto a ponto transmissão com exatamente um remetente e exatamente um receptor às vezes é chamada **unicast**.

Em contraste, em uma rede de transmissão, o canal de comunicação é compartilhado por todas as máquinas da rede; pacotes enviados por qualquer máquina são recebidos por todos os outros. Um campo de endereço dentro de cada pacote especifica o destinatário pretendido. Ao receber um pacote, uma máquina verifica o campo de endereço. Se o pacote estiver dentro tendido para a máquina receptora, essa máquina processa o pacote; se o pacote destina-se a alguma outra máquina, mas é simplesmente ignorado.

Uma rede sem fio é um exemplo comum de um link de transmissão, com comunicação cação compartilhada em uma região de cobertura que depende do canal sem fio e da máquina transmissora. Como uma analogia, considere alguém em uma reunião sala e gritando "Watson, venha aqui. Eu quero você." Embora o pacote possa realmente ser recebido (ouvido) por muitas pessoas, apenas o Watson responderá; os outrosIgnore isso.

Os sistemas de transmissão geralmente também permitem a possibilidade de endereçar um pacote para todos os destinos usando um código especial no campo de endereço. Quando um pacote com este código é transmitido, é recebido e processado por cada máquina na rede-trabalhos. Este modo de operação é chamado de **transmissão**. Alguns sistemas de transmissão também oferecem suporte à transmissão para um subconjunto de máquinas, conhecido como **multicast-ing**.

Um critério alternativo para classificar redes é por escala. A distância é importante como uma métrica de classificação porque diferentes tecnologias são usadas em diferentes escalas diferentes.

Na Fig. 1-6, classificamos vários sistemas de processador por seus aspectos físicos Tamanho. No topo estão as redes de área pessoal, redes destinadas a um pessoa. Além dessas, vêm as redes de longo alcance. Estes podem ser divididos em redes locais, metropolitanas e de área ampla, cada uma em escala crescente. Finalmente, a conexão de duas ou mais redes é chamada de internetwork. O mundo A Internet é certamente o exemplo mais conhecido (mas não o único) de uma internetwork.

Página 42

18

INTRODUÇÃO INDIVÍDUO. 1

Em breve teremos internetworks ainda maiores com a **Internet Interplanetária** que conecta redes através do espaço (Burleigh et al., 2003).

1 m

Metro quadrado

10 m

Quarto

100 m

Construção

Campus

1 km

Cidade

10 km

Interprocessador

distância

Processadores

localizado no mesmo

Exemplo

100 km

País

Continente

1000 km

Planeta

Rede de área pessoal

A Internet

Rede local

Rede de área metropolitana

Rede de longa distância

10.000 km

Figura 1-6. Classificação de processadores interconectados por escala.

Neste livro, vamos nos preocupar com redes em todas essas escalas. No nas seções seguintes, fornecemos uma breve introdução ao hardware de rede por escala.

1.2.1 Redes de área pessoal

PANs (redes de área pessoal) permitem que os dispositivos se comuniquem ao longo do intervalo de uma pessoa. Um exemplo comum é uma rede sem fio que conecta um computador com seus periféricos. Quase todo computador tem um monitor, teclado, mouse e impressora. Sem usar wireless, esta conexão deve ser feita com cabos. Muitos novos usuários têm dificuldade em encontrar os cabos certos e conectá-los colocá-los nos orifícios certos (embora sejam geralmente codificados por cores) que a maioria dos fornecedores de computador oferece a opção de enviar um técnico para a casa do usuário para fazer isso. Para ajudar esses usuários, algumas empresas se reuniram para projetar um curto alcance

rede sem fio chamada **Bluetooth** para conectar esses componentes sem fios. A ideia é que, se seus dispositivos tiverem Bluetooth, você não precisará de cabos. Você apenas coloque-os para baixo, ligue-os e eles funcionam juntos. Para muitas pessoas, essa facilidade de operação é uma grande vantagem.

Na forma mais simples, as redes Bluetooth usam o paradigma mestre-escravo de Fig. 1-7. A unidade do sistema (o PC) é normalmente o mestre, falando com o mouse, teclado, etc., como escravos. O mestre diz aos escravos quais endereços usar, quando eles podem transmitir, por quanto tempo podem transmitir, que frequências podem usar, e assim por diante.

O Bluetooth também pode ser usado em outras configurações. Muitas vezes é usado para conectar um fone de ouvido para um telefone celular sem cabos e pode permitir o seu reproduutor de música digital

Página 43

SEC. 1,2
HARDWARE DE REDE

19

Figura 1-7. Configuração de Bluetooth PAN.

para se conectar ao seu carro apenas sendo colocado dentro do alcance. Um completamente diferente tipo de PAN é formado quando um dispositivo médico integrado, como um marca-passo, bomba de insulina ou aparelho auditivo fala com um controle remoto operado pelo usuário. Nós iremos dis-

Fale sobre o Bluetooth com mais detalhes no capítulo. 4 -

PANs também podem ser construídos com outras tecnologias que se comunicam em curto gamas, como RFID em smartcards e livros de biblioteca. Vamos estudar RFID em Indivíduo. 4 -

1.2.2 Redes locais

O próximo passo é a **LAN** (**Rede Local**). Uma LAN é uma rede privada rede própria que opera dentro e nas proximidades de um único edifício, como uma casa, defice ou fábrica. LANs são amplamente utilizadas para conectar computadores pessoais e consumidores

mais eletrônicos para permitir que eles compartilhem recursos (por exemplo, impressoras) e troquem informações

ção. Quando as LANs são usadas por empresas, elas são chamadas de **redes corporativas** .

LANs sem fio são muito populares hoje em dia, especialmente em residências, escritórios antigos edificios, lanchonetes e outros lugares onde é muito difícil instalar cabos. Nestes sistemas, cada computador tem um modem de rádio e uma antena que ele usa para se comunicar com outros computadores. Na maioria dos casos, cada computador fala a um dispositivo no teto, conforme mostrado na Fig. 1-8 (a). Este dispositivo, chamado de **AP** (**Ponto de acesso**), **roteador sem fio** ou **estação base** , retransmite pacotes entre os computadores sem fio e também entre eles e a Internet. Ser AP é como ser o garoto popular da escola porque todo mundo quer falar com você. Contudo, se outros computadores estiverem próximos o suficiente, eles podem se comunicar diretamente com um

outro em uma configuração ponto a ponto.

Existe um padrão para LANs sem fio chamado **IEEE 802.11** , popularmente conhecido como **WiFi** , que se tornou muito difundido. Ele funciona a velocidades em qualquer lugar de 11

Página 44

20
INTRODUÇÃO
INDIVÍDUO. 1
Ethernet
interruptor
Ports
Para descansar de
rede

Para rede com fio
Acesso
ponto

Figura 1-8. LANs com e sem fio. (a) 802.11. (b) Ethernet comutada.

a centenas de Mbps. (Neste livro, vamos seguir a tradição e medir a linha velocidades em megabits / seg, onde 1 Mbps é 1.000.000 bits / seg, e gigabits / seg, onde 1 Gbps é 1.000.000.000 bits / seg.) Discutiremos 802.11 no cap. 4 - LANs com fio usam uma variedade de tecnologias de transmissão diferentes. O máximo de eles usam fios de cobre, mas alguns usam fibra óptica. LANs são restritos em tamanho, o que significa que o pior caso de tempo de transmissão é limitado e conhecido em advance. Conhecer esses limites ajuda na tarefa de projetar protocolos de rede. Normalmente, as LANs com fio funcionam a velocidades de 100 Mbps a 1 Gbps, têm baixo atraso (microssegundos ou nanossegundos) e cometer poucos erros. LANs mais recentes podem operar erate em até 10 Gbps. Em comparação com as redes sem fio, as LANs com fio os superam em todas as dimensões de desempenho. É mais fácil enviar sinais por um fio ou através de uma fibra do que através do ar.

A topologia de muitas LANs com fio é construída a partir de links ponto a ponto. IEEE 802.3, popularmente chamado de **Ethernet**, é, de longe, o tipo mais comum de conexão com fio LAN. A Figura 1-8 (b) mostra um exemplo de topologia de **Ethernet comutada**. Cada computador fala o protocolo Ethernet e se conecta a uma caixa chamada **switch** com um link ponto a ponto. Daí o nome. Um switch tem várias **portas**, cada uma das quais pode se conectar a um computador. O trabalho do switch é retransmitir pacotes entre computadores que estão conectados a ele, usando o endereço em cada pacote para determinar para qual computador enviar.

Para construir LANs maiores, os switches podem ser conectados uns aos outros usando seus portas. O que acontece se você conectá-los em um loop? A rede ainda trabalhos? Felizmente, os designers pensaram neste caso. É função do protocolo decidir quais caminhos os pacotes devem viajar para chegar com segurança ao computador pretendido.

Veremos como isso funciona no cap. 4 -

Também é possível dividir uma grande LAN física em duas lógicas menores LANs. Você pode se perguntar por que isso seria útil. Às vezes, o layout do equipamento de rede não corresponde à estrutura da organização. Por exemplo, o

Página 45

SEC. 1,2
HARDWARE DE REDE

21

departamentos de engenharia e finanças de uma empresa podem ter computadores no mesmo LAN física porque eles estão na mesma ala do prédio, mas pode ser mais fácil gerenciar o sistema se a engenharia e as finanças logicamente cada uma tivesse sua própria rede **Virtual LAN** ou **VLAN**. Neste projeto, cada porta é marcada com um "cor," diz verde para engenharia e vermelho para finanças. O interruptor então avança pacotes de modo que os computadores conectados às portas verdes sejam separados dos computadores ligados às portas vermelhas. Pacotes de difusão enviados em uma porta vermelha, por exemplo,

não será recebido em uma porta verde, como se houvesse dois

LANs. Abordaremos as VLANs no final do capítulo. 4 -

Existem outras topologias de LAN com fio também. Na verdade, a Ethernet comutada é uma versão moderna do design Ethernet original que transmite todos os pacotes em único cabo linear. No máximo uma máquina pode transmitir com sucesso por vez, e um mecanismo de arbitragem distribuída foi usado para resolver conflitos. Usou um algoritmo simples: os computadores podem transmitir sempre que o cabo estiver ocioso. Se dois ou mais pacotes colidiram, cada computador apenas esperou um tempo aleatório e tentou mais tarde. Chamaremos essa versão de **Ethernet clássica** para maior clareza e, como você suspeitou, você aprenderá sobre isso no cap. 4 -

Ambas as redes de transmissão sem fio e com fio podem ser divididas em estáticas e designs dinâmicos, dependendo de como o canal é alocado. Um típico estático al-

localização seria dividir o tempo em intervalos discretos e usar um round-robin algoritmo, permitindo que cada máquina transmita apenas quando seu intervalo de tempo chegar. A alocação estática desperdiça a capacidade do canal quando uma máquina não tem nada a dizer durante

o slot alocado, então a maioria dos sistemas tenta alocar o canal dinamicamente (ou seja, sob demanda).

Os métodos de alocação dinâmica para um canal comum são centralizados ou descentralizado. No método de alocação de canal centralizado, há um único entidade, por exemplo, a estação base em redes celulares, que determina quem vai Próximo. Ele pode fazer isso aceitando vários pacotes e priorizando-os de acordo com a algum algoritmo interno. No método de alocação de canal descentralizado, não há entidade central; cada máquina deve decidir por si mesma se deve transmitir.

Você pode pensar que essa abordagem levaria ao caos, mas não leva. Depois nós estudará muitos algoritmos projetados para trazer ordem ao caos potencial.

Vale a pena gastar um pouco mais de tempo discutindo as LANs em casa. No futuro, é provável que todos os eletrodomésticos da casa sejam capazes de comunicar com todos os outros aparelhos, e todos eles estarão acessíveis no Internet. É provável que este desenvolvimento seja um daqueles conceitos visionários que ninguém pediu (como controles remotos de TV ou telefones celulares), mas uma vez ninguém pode imaginar como eles viveram sem eles.

Muitos dispositivos já podem ser conectados em rede. Estes incluem computadores, dispositivos de entretenimento, como TVs e DVDs, telefones e outros eletrônicos, como câmeras, aparelhos como rádios-relógios e infraestrutura como medidores de utilidade e termostatos. Essa tendência só vai continuar. Por exemplo, o casa média provavelmente tem uma dúzia de relógios (por exemplo, em eletrodomésticos), todos os quais poderiam

Página 46

22

INTRODUÇÃO

INDIVÍDUO. 1

ajustar para o horário de verão automaticamente se os relógios estiverem na Internet.

O monitoramento remoto da casa é um provável vencedor, como muitos filhos adultos fariam estar disposto a gastar algum dinheiro para ajudar seus pais idosos a viver com segurança em suas próprias casas.

Embora possamos pensar na rede doméstica apenas como outra LAN, é mais provavelmente terá propriedades diferentes das outras redes. Primeiro, os de vícios devem ser muito fáceis de instalar. Roteadores sem fio são os que mais retornam item eletrônico sumer. As pessoas compram um porque querem uma rede sem fio em casa, descubra que não funciona "fora da caixa" e, em seguida, devolva-o em vez de ouvir música de elevador enquanto espera na linha de apoio técnico.

Em segundo lugar, a rede e os dispositivos devem funcionar à prova de falhas. Ar condicionado os dicionadores costumavam ter um botão com quatro configurações: OFF, LOW, MEDIUM e ALTO. Agora eles têm manuais de 30 páginas. Assim que estiverem em rede, espere o capítulo sobre segurança sozinho com 30 páginas. Este é um problema porque apenas com os usuários de computador estão acostumados a tolerar produtos que não funcionam; o público que compra televisão e geladeira é muito menos tolerante. Eles esperam produtos para funcionar 100% sem a necessidade de contratar geek.

Terceiro, o preço baixo é essencial para o sucesso. As pessoas não pagarão um prêmio de \$ 50 para um termostato de Internet, porque poucas pessoas consideram monitorar a temperatura de suas casas

peratura do trabalho tão importante. Por US \$ 5 extras, porém, pode vender.

Quarto, deve ser possível começar com um ou dois dispositivos e expandir o alcance da rede gradualmente. Isso significa nenhuma guerra de formato. Dizendo aos consumidores comprar periféricos com interfaces IEEE 1394 (FireWire) e alguns anos depois retrair isso e dizer que USB 2.0 é a interface do mês e depois mudar fazer isso para 802.11g — opa, não, faça isso 802.11n — quero dizer 802.16 (fio diferente

menos redes) - deixará os consumidores muito nervosos. A interface de rede terá que permanecer estável por décadas, como os padrões de transmissão de televisão. Quinto, a segurança e a confiabilidade serão muito importantes. Perder alguns arquivos para um vírus de e-mail é uma coisa; tendo um ladrão desarmado seu sistema de segurança de seu computador móvel e saquear sua casa é algo bem diferente.

Uma questão interessante é se as redes domésticas serão com ou sem fio.

A conveniência e o custo favorecem a rede sem fio porque não há fios para ajuste, ou pior, retrofit. A segurança favorece a rede com fio porque as ondas de rádio que as redes sem fio usam são muito boas para atravessar paredes. Nem todo mundo é muito feliz com a ideia de ter os vizinhos pegando carona em sua Internet conexão e lendo seu e-mail. No cap. 8 vamos estudar como a criptografia pode ser usado para fornecer segurança, mas é mais fácil falar do que fazer com inexperientes Comercial.

Uma terceira opção que pode ser atraente é reutilizar as redes que já estão em casa. O candidato óbvio são os fios elétricos que são instalados em toda a casa. **Redes de energia** permitem que dispositivos que se conectam a tomadas transmitir informações por toda a casa. Você tem que conectar a TV de qualquer maneira, e desta forma ele pode obter conectividade com a Internet ao mesmo tempo. A dificuldade é

Página 47

SEC. 1,2

HARDWARE DE REDE

23

como transportar sinais de energia e dados ao mesmo tempo. Parte da resposta é que usam bandas de frequência diferentes.

Resumindo, as LANs domésticas oferecem muitas oportunidades e desafios. A maioria dos o último está relacionado à necessidade de as redes serem fáceis de gerenciar, confiáveis e seguro, especialmente nas mãos de usuários não técnicos, bem como de baixo custo.

1.2.3 Redes de Área Metropolitana

A MAN (**M**etropolitan **A**rea **N**etwork) cobre uma cidade. O ex-mais conhecido muitos dos MANs são as redes de televisão a cabo disponíveis em muitas cidades. Esses sistemas cresceram a partir de sistemas de antenas comunitárias anteriores usados em áreas com

má recepção de televisão over-the-air. Naqueles primeiros sistemas, uma grande antena era colocada no topo de uma colina próxima e um sinal foi então canalizado para os assinantes casas.

No início, esses sistemas eram projetados localmente, ad hoc. Então as empresas começaram entrando no negócio, conseguindo contratos dos governos locais para conectar cidades de pneus. A próxima etapa foi a programação da televisão e até mesmo canais inteiros projetado apenas para cabo. Muitas vezes, esses canais eram altamente especializados, como todos notícias, todos os esportes, toda culinária, toda jardinagem e assim por diante. Mas desde o início até o final da década de 1990, eles se destinavam apenas à recepção de televisão.

Quando a Internet começou a atrair grande audiência, a rede de TV a cabo os operadores começaram a perceber que, com algumas mudanças no sistema, eles poderiam vide serviço de Internet bidirecional em partes não utilizadas do espectro. Nesse ponto, o sistema de TV a cabo começou a se transformar de simplesmente uma forma de distribuir televisão para um rede de área metropolitana. Para uma primeira aproximação, um MAN pode parecer um pouco algo como o sistema mostrado na Fig. 1-9. Nesta figura, vemos ambos os sinais de televisão nais e Internet sendo alimentados no **headend de cabo** centralizado para posterior descontinuação tributação às casas das pessoas. Voltaremos a esse assunto em detalhes no Cap.

2

A televisão a cabo não é a única MAN. Desenvolvimentos recentes em alta o acesso à Internet sem fio de velocidade resultou em outro MAN, que foi padronizado como IEEE 802.16 e é popularmente conhecido como **WiMAX** . Vamos olhar nele no cap. 4 -

1.2.4 Redes de longa distância

Uma WAN (Wide Area Network) abrange uma grande área geográfica, muitas vezes um país ou continente. Começaremos nossa discussão com WANs com fio, usando o exemplo de empresa com filiais em diferentes cidades.

A WAN na Figura 1-10 é uma rede que conecta escritórios em Perth, Melbourne, e Brisbane. Cada um desses escritórios contém computadores destinados ao usuário em execução (ou seja, aplicativos). Seguiremos o uso tradicional e chamá-los de **hospedeiros**. O resto da rede que conecta esses hosts é então chamado de

Página 48

24

INTRODUÇÃO INDIVÍDUO. 1

Internet
Antena
Junção
caixa
Extremidade principal

Figura 1-9. Uma rede de área metropolitana baseada em TV a cabo.

sub-rede de comunicação, ou simplesmente **sub - rede**. O trabalho da sub-rede é transportar mensagens de host para host, assim como o sistema telefônico carrega palavras (na verdade, apenas sons) do alto-falante para o ouvinte.

Na maioria das WANs, a sub-rede consiste em dois componentes distintos: transmissão linhas e elementos de comutação. As **linhas de transmissão** movem bits entre as máquinas. Eles podem ser feitos de fio de cobre, fibra óptica ou mesmo links de rádio. Mais com as empresas não têm linhas de transmissão espalhadas, então, em vez disso, alugam as linhas de uma empresa de telecomunicações. **Elementos de troca**, ou apenas **interruptores**, são computadores especializados que conectam duas ou mais linhas de transmissão. Quando dados chegar em uma linha de entrada, o elemento de comutação deve escolher uma linha de saída em qual encaminhá-los. Esses computadores de comutação foram chamados por vários nomes no passado; o nome **roteador** agora é o mais comumente usado. Infelizmente, algumas pessoas pronunciam-no como "rooter", enquanto outras rimam com "duvidador".

A determinação da pronúncia correta será deixada como um exercício para o leitor.

(Observação: a resposta correta percebida pode depender de onde você mora.)

Um breve comentário sobre o termo "sub-rede" está em ordem aqui. Originalmente, é **o único** significado era a coleção de roteadores e linhas de comunicação que se moviam pacotes do host de origem ao host de destino. Os leitores devem estar cientes de que adquiriu um segundo significado mais recente em conjunto com o anúncio de rede vestir. Discutiremos esse significado no cap. 5 e fique com o original significado (uma coleção de linhas e roteadores) até então.

A WAN, conforme descrevemos, é semelhante a uma grande LAN com fio, mas existem algumas diferenças importantes que vão além dos fios longos. Normalmente em um WAN, os hosts e a sub-rede pertencem e são operados por pessoas diferentes. Na nossa

Página 49

SEC. 1,2 HARDWARE DE REDE

25

Sub-rede
Roteador
Perth
Brisbane
Melbourne
Transmissão
linha

Figura 1-10. WAN que conecta três filiais na Austrália.

por exemplo, os funcionários podem ser responsáveis por seus próprios computadores, enquanto o Departamento de TI da empresa é responsável pelo restante da rede. Vamos ver limites mais claros nos próximos exemplos, em que o provedor de rede ou a companhia telefônica opera a sub-rede. Separação da comunicação pura aspectos da rede (a sub-rede) dos aspectos do aplicativo (os hosts) muito simplifica o design geral da rede.

Uma segunda diferença é que os roteadores geralmente conectam diferentes tipos de tecnologia de rede. As redes dentro dos escritórios podem ser comutadas Ether-rede, por exemplo, enquanto as linhas de transmissão de longa distância podem ser links SONET (que abordaremos no Capítulo 2). Algum dispositivo precisa se juntar a eles. O astuto leitor perceberá que isso vai além de nossa definição de rede. Isso significa que muitas WANs serão de fato **internetworks**, ou redes compostas que são composta por mais de uma rede. Teremos mais a dizer sobre a internet-trabalha na próxima seção.

Uma diferença final está no que está conectado à sub-rede. Isso pode ser individual computadores duplos, como era o caso da conexão a LANs, ou poderia ser inteiro LANs. É assim que redes maiores são construídas a partir de redes menores. Tanto quanto o sub-net está em causa, ele faz o mesmo trabalho.

Agora estamos em posição de examinar duas outras variedades de WANs. Em primeiro lugar do que alugar linhas de transmissão dedicadas, uma empresa pode conectar seus escritórios ao Internet. Isso permite que as conexões sejam feitas entre os escritórios como links virtuais

Página 50

26

INTRODUÇÃO INDIVÍDUO. 1

que usam a capacidade subjacente da Internet. Este arranjo, mostrado em Figura 1-11 é chamada de **VPN** (**Virtual Private Network**). Comparado com o dedi-acordo, uma VPN tem a vantagem usual da virtualização, que é que ele fornece reutilização flexível de um recurso (conectividade com a Internet). Considere como é fácil

é adicionar um quarto escritório para ver isso. Uma VPN também tem a desvantagem usual de virtualização, que é uma falta de controle sobre os recursos subjacentes. Com um linha dedicada, a capacidade é clara. Com uma VPN, sua milhagem pode variar com seu serviço de Internet.

Internet
Perth
Brisbane
Melbourne
Link através do
Internet

Figura 1-11. WAN usando uma rede privada virtual.

A segunda variação é que a sub-rede pode ser administrada por uma empresa diferente. O operador de sub-rede é conhecido como **provedor de serviços de rede** e os escritórios são seus clientes. Essa estrutura é mostrada na Figura 1-12. O operador de sub-rede irá conectar-se a outros clientes também, desde que eles possam pagar e fornecer serviços. Uma vez que seria um serviço de rede decepcionante se os clientes pudessem apenas enviar pacotes um para o outro, o operador de sub-rede também se conectará a outras redes que fazem parte da Internet. Esse operador de sub-rede é chamado de **ISP** (**Internet Provedor de Serviços**) e a sub-rede é uma **rede ISP**. Seus clientes que se conectam para o ISP receber serviço de Internet.

Podemos usar a rede do ISP para visualizar algumas questões-chave que estudaremos capítulos posteriores. Na maioria das WANs, a rede contém muitas linhas de transmissão, cada um conectando um par de roteadores. Se dois roteadores que não compartilham uma transmissão deseja se comunicar, eles devem fazer isso indiretamente, através de outros roteadores. Lá

Página 51

SEC. 1,2 HARDWARE DE REDE

27

Rede ISP
Perth
Brisbane
Melbourne
Transmissão

linha
Cliente
rede

Figura 1-12. WAN usando uma rede ISP.

pode haver muitos caminhos na rede que conectam esses dois roteadores. Como a rede-trabalho toma a decisão de qual caminho usar é chamado de **algoritmo de roteamento**. Muitos desses algoritmos existem. Como cada roteador toma a decisão de onde enviar um pacote em seguida é chamado de **algoritmo de encaminhamento**. Muitos deles também existem.

Estudaremos alguns dos dois tipos em detalhes no Cap. 5. Outros tipos de WANs fazem uso intenso de tecnologias sem fio. No satélite sistemas, cada computador no solo tem uma antena através da qual pode enviar dados para e receber dados de um satélite em órbita. Todos os computadores podem ouvir o saída do satélite e, em alguns casos, eles também podem ouvir o transmissões de seus colegas computadores para o satélite também. Redes de satélite são inherentemente transmitidos e são mais úteis quando a propriedade de transmissão é importante.

A rede de telefonia celular é outro exemplo de WAN que usa fios menos tecnologia. Este sistema já passou por três gerações e um o quarto está no horizonte. A primeira geração era analógica e apenas para voz. A segunda geração era digital e apenas para voz. A terceira geração é digital e é para voz e dados. Cada estação base celular cobre uma distância muito maior do que uma LAN sem fio, com um alcance medido em quilômetros em vez de dezenas de metros. As estações base são conectadas entre si por uma rede de backbone trabalho que geralmente é conectado. As taxas de dados das redes celulares costumam estar na ordem de 1 Mbps, muito menor do que uma LAN sem fio que pode variar até na ordem de 100 Mbps. Teremos muito a dizer sobre essas redes no cap. 2

Página 52

28

INTRODUÇÃO
INDIVÍDUO. 1

1.2.5 Internetworks

Existem muitas redes no mundo, geralmente com hardware e software diferentes. Pessoas conectadas a uma rede geralmente desejam se comunicar com pessoas conectadas para um diferente. A realização deste desejo requer que diferente, e frequentemente incompatíveis, as redes devem ser conectadas. Uma coleção de rede interconectada funciona é chamado de **internetwork** ou **internet**. Esses termos serão usados em uma geração sentido erico, em contraste com a Internet mundial (que é uma internet específica), que sempre capitalizaremos. A Internet usa redes ISP para conectar-se redes empresariais, redes domésticas e muitas outras redes. Vamos olhar para o Internet em grande detalhe posteriormente neste livro.

Sub-redes, redes e internetworks são freqüentemente confundidas. O termo " sub-rede " faz mais sentido no contexto de uma rede de longa distância, onde se refere ao coleção de roteadores e linhas de comunicação de propriedade da operadora de rede. Como por analogia, o sistema telefônico consiste em centrais telefônicas de comutação conectadas ligados uns aos outros por linhas de alta velocidade e para casas e empresas por linhas de baixa velocidade

linhas. Essas linhas e equipamentos, pertencentes e gerenciados pela empresa telefônica empresa, forma a sub-rede do sistema telefônico. Os próprios telefones (o hosts nesta analogia) não fazem parte da sub-rede.

Uma rede é formada pela combinação de uma sub-rede e seus hosts. Contudo, a palavra " rede " também é frequentemente usada em um sentido vago. Uma sub-rede pode ser descrita como uma rede, como no caso da " rede ISP " da Figura 1.12. Um interrede também pode ser descrita como uma rede, como no caso da WAN em Fig. 1-10. Seguiremos uma prática semelhante, e se estivermos distinguindo uma rede de outros arranjos, manteremos nossa definição original de coleção de computadores interligados por uma única tecnologia.

Vamos dizer mais sobre o que constitui uma internetwork. Nós sabemos que um a ternet é formada quando redes distintas são interconectadas. Em nossa opinião, conectar conectar uma LAN e uma WAN ou conectar duas LANs é a maneira usual de formar uma interface rede, mas há pouco acordo na indústria sobre a terminologia nesta área.

Existem duas regras práticas que são úteis. Primeiro, se diferentes organizações têm pagos para construir diferentes partes da rede e cada uma mantém sua parte, nós ter uma internetwork em vez de uma única rede. Em segundo lugar, se a tecnologia subjacente tecnologia é diferente em diferentes partes (por exemplo, transmissão versus ponto a ponto e com fio versus sem fio), provavelmente temos uma internetwork.

Para ir mais fundo, precisamos falar sobre como duas redes diferentes podem ser conectadas. O nome geral de uma máquina que faz uma conexão entre dois ou mais redes é fornecido a tradução necessária, tanto em termos de hardware e software, é um **portal**. Gateways são diferenciados pela camada em que eles operam na hierarquia de protocolo. Teremos muito mais a dizer sobre leigos hierarquias de protocolo e hierarquias começando na próxima seção, mas por enquanto imagine que camadas superiores são mais vinculadas a aplicativos, como a Web, e camadas inferiores são mais vinculadas a links de transmissão, como Ethernet.

Página 53

SEC. 1,2
HARDWARE DE REDE

29

Uma vez que o benefício de formar uma internet é conectar computadores através da rede funciona, não queremos usar um gateway de nível muito baixo ou não poderemos fazer conexões entre diferentes tipos de redes. Não queremos usar um gateway de nível muito alto ou a conexão só funcionará para um aplicativo particular. O nível no meio que está "certo" é geralmente chamado de rede camada, e um roteador é um gateway que alterna pacotes na camada de rede. Nós agora pode detectar uma Internet encontrando uma rede que tenha roteadores.

1.3 SOFTWARE DE REDE

As primeiras redes de computadores foram projetadas com o hardware como principal preocupação e o software como uma reflexão tardia. Essa estratégia não funciona mais. Internet - o software de trabalho agora é altamente estruturado. Nas seções a seguir, examinaremos a técnica de estruturação de software em alguns detalhes. A abordagem descrita aqui forma a pedra angular de todo o livro e ocorrerá repetidamente mais tarde.

1.3.1 Hierarquias de protocolo

Para reduzir sua complexidade de design, a maioria das redes são organizadas como uma pilha de **camadas** ou **níveis**, cada um construído sobre o que está abaixo dele. O número de camadas, o nome de cada camada, o conteúdo de cada camada e a função de cada camada diferem de rede para rede. O objetivo de cada camada é oferecer certos serviços para as camadas superiores, protegendo essas camadas dos detalhes de como os serviços oferecidos são realmente implementados. Em certo sentido, cada camada é uma espécie de vir-

máquina virtual, oferecendo certos serviços para a camada acima dela.

Este conceito é realmente familiar e é usado em toda a ciência da computação, onde é conhecido como ocultação de informações, tipos de dados abstratos, dados encapsulamento e programação orientada a objetos. A ideia fundamental é que um determinado software (ou hardware) fornece um serviço aos usuários, mas mantém os detalhes de seu estado interno e algoritmos ocultos deles.

Quando a camada n em uma máquina mantém uma conversa com a camada n em outra máquina, as regras e convenções usadas nesta conversa são coletivamente conhecido como protocolo da camada n . Basicamente, um **protocolo** é um acordo entre o comunicar às partes como a comunicação deve proceder. Como analogia, quando uma mulher é apresentada a um homem, ela pode escolher estender a mão. Ele, por sua vez, pode decidir sacudi-lo ou beijá-lo, dependendo, por exemplo, de onde er ela é uma advogada americana em uma reunião de negócios ou uma princesa europeia em uma bola formal. A violação do protocolo tornará a comunicação mais difícil, se

não completamente impossível.

Uma rede de cinco camadas é ilustrada na Figura 1-13. As entidades que compõem o as camadas correspondentes em máquinas diferentes são chamadas de **pares**. Os pares podem ser

Página 54

30

INTRODUÇÃO

INDIVÍDUO. 1

processos de software, dispositivos de hardware ou até mesmo seres humanos. Em outras palavras, é os pares que se comunicam usando o protocolo para se comunicarem.

Camada 5
Camada 4
Camada 3
Camada 2
Camada 1
Host 1
Interface da camada 4/5
Interface da camada 3/4
Interface da camada 2/3
Interface da camada 1/2
Protocolo da camada 5
Camada 5
Camada 4
Camada 3
Camada 2
Camada 1
Host 2
Protocolo da camada 4
Protocolo da camada 3
Protocolo da camada 2
Protocolo da camada 1
Meio físico

Figura 1-13. Camadas, protocolos e interfaces.

Na realidade, nenhum dado é transferido diretamente da camada *n* em uma máquina para camada *n* em outra máquina. Em vez disso, cada camada passa dados e informações de controle mação para a camada imediatamente abaixo dela, até que a camada mais baixa seja alcançada. Abaixo

a camada 1 é o **meio físico** por **meio do** qual ocorre a comunicação real. No Fig. 1-13, a comunicação virtual é mostrada por linhas pontilhadas e comunicação física cátion por linhas sólidas.

Entre cada par de camadas adjacentes existe uma **interface**. A interface define quais operações e serviços primitivos a camada inferior disponibiliza para o superior. Quando os designers de rede decidem quantas camadas incluir em uma rede trabalho e o que cada um deve fazer, uma das considerações mais importantes é definindo interfaces limpas entre as camadas. Fazer isso, por sua vez, requer que cada camada execute uma coleção específica de funções bem compreendidas. Além de minimizando a quantidade de informações que devem ser passadas entre as camadas, interfaces de corte também tornam mais simples substituir uma camada por uma completamente diferente

protocolo ou implementação (por exemplo, substituição de todas as linhas telefônicas por satélite canais) porque tudo o que é exigido do novo protocolo ou implementação é que oferece exatamente o mesmo conjunto de serviços para o vizinho de cima que o antigo fez. É comum que diferentes hosts usem diferentes implementações do mesmo protocolo (geralmente escrito por empresas diferentes). Na verdade, o próprio protocolo pode mudar em alguma camada sem que as camadas acima e abaixo dela percebam.

Página 55

SEC. 1,3

SOFTWARE DE REDE

31

Um conjunto de camadas e protocolos é chamado de **arquitetura de rede**. A especificação icação de uma arquitetura deve conter informações suficientes para permitir uma implementação mentor para escrever o programa ou construir o hardware para cada camada de modo que obedecer corretamente ao protocolo apropriado. Nem os detalhes da implementação

nem a especificação das interfaces faz parte da arquitetura porque estas são escondido dentro das máquinas e não visível do lado de fora. Não é mesmo necessário que as interfaces em todas as máquinas de uma rede sejam as mesmas, desde que que cada máquina pode usar corretamente todos os protocolos. Uma lista dos protocolos usados por um determinado sistema, um protocolo por camada é chamado de **pilha de protocolos**. Rede arquiteturas, pilhas de protocolo e os próprios protocolos são os principais subjects deste livro.

Uma analogia pode ajudar a explicar a ideia de comunicação multicamadas. Imagine dois filósofos (processos de pares na camada 3), um dos quais fala Urdu e Inglês e um deles fala chinês e francês. Uma vez que eles não têm idioma único, cada um deles envolve um tradutor (processos de pares na camada 2), cada um dos quem por sua vez contata uma secretária (processos de pares na camada 1). Filósofo 1 deseja transmitir sua afeição por *oryctolagus cuniculus* ao seu par. Para fazer isso, ele passa uma mensagem (em inglês) pela interface 2/3 para seu tradutor, dizendo " Eu como coelhos ", conforme ilustrado na Figura 1-14. Os tradutores concordaram em um neutro idioma conhecido por ambos, holandês, então a mensagem é convertida para " Ik vind konijnen leuk. " A escolha do idioma é o protocolo da camada 2 e fica a cargo do processos pares da camada 2.

O tradutor então passa a mensagem para uma secretária para transmissão, por exemplo, por e-mail (o protocolo da camada 1). Quando a mensagem chega ao outro secretário, é passado para o tradutor local, que o traduz para o francês e passa-o através da interface 2/3 para o segundo filósofo. Observe que cada protocolo é completamente independente das outras, desde que as interfaces não sejam mudou. Os tradutores podem mudar de holandês para, digamos, finlandês, à vontade, desde que que ambos concordam e nenhum muda sua interface com a camada 1 ou camada 3. Da mesma forma, as secretárias podem mudar do e-mail para o telefone sem perturbar ing (ou mesmo informar) as outras camadas. Cada processo pode adicionar algumas informações destinado apenas para seu par. Essa informação não é passada para a camada acima.

Agora considere um exemplo mais técnico: como fornecer comunicação para a camada superior da rede de cinco camadas da Figura 1.15. Uma mensagem, M , é produzida por um processo de aplicação executado na camada 5 e fornecido à camada 4 para transmissão. A camada 4 coloca um **cabeçalho** na frente da mensagem para identificar a mensagem e passa o resultado para a camada 3. O cabeçalho inclui informações de controle, como endereços, para permitir que a camada 4 na máquina de destino entregue a mensagem. Outro exemplos de informações de controle usados em algumas camadas são números de sequência (no caso a camada inferior não preserva a ordem, os tamanhos e os horários das mensagens. Em muitas redes, nenhum limite é colocado no tamanho das mensagens transmitidas em o protocolo da camada 4, mas quase sempre há um limite imposto pelo protocolo da camada 3 toco. Consequentemente, a camada 3 deve dividir as mensagens recebidas em menores

32

INTRODUÇÃO INDIVÍDUO. 1

Eu gosto

coelhos

Localização A

3

2

1

3

2

1

Localização B

mensagem

Filósofo

Tradutor

secretário

Em formação

para o remoto

tradutor

Em formação

para o remoto

secretário

```

L: holandês
Ik vind
Konijnen
Leuk
Fax # ---
L: holandês
Ik vind
Konijnen
Leuk
J'aime
bien les
lapins
L: holandês
Ik vind
Konijnen
Leuk
Fax # ---
L: holandês
Ik vind
Konijnen
Leuk

```

Figura 1-14. A arquitetura filósofo-tradutor-secretário.

unidades, pacotes, prefixando um cabeçalho da camada 3 para cada pacote. Neste exemplo, M é dividida em duas partes, M_1 e M_2 , que serão transmitidos separadamente.

A camada 3 decide qual das linhas de saída usar e passa os pacotes para camada 2. A camada 2 adiciona a cada peça não apenas um cabeçalho, mas também um trailer, e fornece

a unidade resultante para a camada 1 para transmissão física. Na máquina receptora a mensagem se move para cima, de camada em camada, com cabeçalhos sendo removidos conforme ele progride. Nenhum dos cabeçalhos das camadas abaixo de n é passado para a camada n .

O importante a entender sobre a Fig. 1-15 é a relação entre o comunicação virtual e real e a diferença entre protocolos e interrostos. Os processos de pares na camada 4, por exemplo, pensam conceitualmente em seus comunicação como sendo "horizontal", usando o protocolo da camada 4. Cada um é provavelmente terá procedimentos chamados de algo como *SendToOtherSide* e *GetFrom-Outro Lado*, embora esses procedimentos realmente se comuniquem com as camadas inferiores na interface 3/4, e não com o outro lado.

SEC. 1,3 SOFTWARE DE REDE

33

```

H2 H3 H4
M1
T2
H2 H3
H2
T2
H2 H3 H4
M1
T2
H2 H3
H2
T2
H3 H4
M1
H3
H2
H3 H4
M1
H3
H2
H4
M
H4
M
M
M
Camada 2
protocolo
2
Camada 3
protocolo
Protocolo da camada 4
Protocolo da camada 5

```

3
4
5
1
Camada
Máquina fonte
Máquina de destino

Figura 1-15. Fluxo de informações de exemplo que apóia a comunicação virtual em camada 5.

A abstração do processo ponto a ponto é crucial para todo projeto de rede. Usando-o, o tarefa incontrolável de projetar a rede completa pode ser dividida em vários problemas de design menores e gerenciáveis, ou seja, o design das camadas individuais. Embora o Sec. 1.3 é chamado de " Software de Rede ", vale a pena ressaltar que as camadas inferiores de uma hierarquia de protocolo são frequentemente implementadas em hardware

ou firmware. No entanto, algoritmos de protocolo complexos estão envolvidos, mesmo se eles estão embutidos (no todo ou em parte) no hardware.

1.3.2 Problemas de projeto para as camadas

Alguns dos principais problemas de design que ocorrem em redes de computadores surgirão camada após camada. Abaixo, iremos mencionar brevemente os mais importantes.

Confiabilidade é a questão do projeto de fazer uma rede que opere corretamente mesmo que seja feito de uma coleção de componentes que são eles próprios não confiável. Pense nos bits de um pacote viajando pela rede. Lá é uma chance de que alguns desses bits sejam recebidos danificados (invertidos) devido a ruído elétrico acidental, sinais sem fio aleatórios, falhas de hardware, bugs de software e em breve. Como é possível encontrar e corrigir esses erros?

Um mecanismo para encontrar erros em informações recebidas usa códigos para **er-detecção de ror**. As informações recebidas incorretamente podem ser retransmitidas

Página 58

34

INTRODUÇÃO
INDIVÍDUO. 1

até que seja recebido corretamente. Códigos mais poderosos permitem a **correção de erros**, onde a mensagem correta é recuperada dos bits possivelmente incorretos que foram recebido originalmente. Ambos os mecanismos funcionam adicionando informações redundantes mação. Eles são usados em camadas baixas, para proteger os pacotes enviados por links individuais, e camadas altas, para verificar se o conteúdo correto foi recebido.

Outro problema de confiabilidade é encontrar um caminho de trabalho em uma rede. Frequentemente

existem vários caminhos entre uma origem e um destino, e em uma grande rede, pode haver alguns links ou roteadores quebrados. Suponha que a rede seja na Alemanha. Os pacotes enviados de Londres para Roma via Alemanha não receberão , mas poderíamos, em vez disso, enviar pacotes de Londres para Roma via Paris. o a rede deve tomar essa decisão automaticamente. Este tópico é chamado de **roteamento** .

Uma segunda questão de design diz respeito à evolução da rede. Com o tempo, as obras crescem e surgem novos designs que precisam ser conectados ao rede ing. Vimos recentemente o mecanismo de estruturação chave usado para suprir mudança de porta dividindo o problema geral e ocultando detalhes de implementação: **camadas de protocolo** . Existem muitas outras estratégias também.

Uma vez que há muitos computadores na rede, cada camada precisa de um mecanismo para identificar os remetentes e receptores que estão envolvidos em uma mensagem particular sábio. Este mecanismo é chamado de **endereçamento** ou **nomenclatura** , no nível inferior e superior ers, respectivamente.

Um aspecto do crescimento é que diferentes tecnologias de rede costumam ter limitações diferentes. Por exemplo, nem todos os canais de comunicação preservam o ordem das mensagens enviadas neles, levando a soluções que numeram as mensagens. A- outro exemplo são as diferenças no tamanho máximo de uma mensagem que as redes pode transmitir. Isso leva a mecanismos para desmontar, transmitir e então

remontagem de mensagens. Este tópico geral é chamado de **internetworking**. Quando as redes ficam grandes, surgem novos problemas. As cidades podem ter engarrafamentos, um

falta de números de telefone e é fácil se perder. Muitas pessoas não têm esses problemas em seu próprio bairro, mas em toda a cidade, podem ser um grande problema. Projetos que continuam a funcionar bem quando a rede fica grande são considerados **escalável**.

Uma terceira questão de design é a alocação de recursos. As redes fornecem um serviço para hosts de seus recursos subjacentes, como a capacidade das linhas de transmissão.

Para fazer isso bem, eles precisam de mecanismos que dividam seus recursos para que um host não interfere muito com o outro.

Muitos projetos compartilham a largura de banda da rede dinamicamente, de acordo com o necessário de prazo dos hosts, em vez de dar a cada host uma fração fixa da banda largura que pode ou não usar. Este projeto é chamado de **multiplexação estatística**, significando compartilhamento com base nas estatísticas de demanda. Pode ser aplicado em camadas baixas

para um único link, ou em camadas altas para uma rede ou mesmo aplicativos que usam a rede.

Um problema de alocação que ocorre em todos os níveis é como manter um remetente rápido de inundar um receptor lento com dados. Feedback do receptor para o

Página 59

SEC. 1,3
SOFTWARE DE REDE

35

remetente é freqüentemente usado. Este assunto é chamado de **controle de fluxo**. Às vezes o problema é que a rede está sobrecarregada porque muitos computadores querem enviar muito tráfego e a rede não pode entregar tudo. Essa sobrecarga da rede é chamada de **congestionamento**. Uma estratégia é para cada computador reduzir seu demanda quando experimenta congestionamento. Ele também pode ser usado em todas as camadas. É interessante observar que a rede tem mais recursos a oferecer do que simplesmente largura de banda. Para usos como o transporte de vídeo ao vivo, a oportunidade de entrega

importa muito. A maioria das redes deve fornecer serviço para aplicativos que desejam essa entrega em **tempo real** ao mesmo tempo em que prestam serviço aos aplicativos que desejam alto rendimento. **Qualidade de serviço** é o nome dado aos mecanismos que conciliar essas demandas concorrentes.

O último grande problema de design é proteger a rede, defendendo-a contra diferentes tipos de ameaças. Uma das ameaças que mencionamos anteriormente é que de espionar comunicações. Mecanismos que fornecem **confidencialidade** defender contra essa ameaça e eles são usados em várias camadas. Mecanismos para **autenticação** impede que alguém se faça passar por outra pessoa. Eles podem ser usado para diferenciar sites de bancos falsos do verdadeiro, ou para permitir que a rede de celular verifique se uma chamada realmente está vindo de seu telefone para que você pague a conta. Outros mecanismos de **integridade** evitam mudanças sub-reptícias nas mensagens, como alterar "debitar \$ 10 da minha conta" para "debitar \$ 1000 da minha conta". Todos os esses projetos são baseados na criptografia, que estudaremos no Cap. 8

1.3.3 Serviço Orientado a Conexão Versus Sem Conexão

As camadas podem oferecer dois tipos diferentes de serviço às camadas acima delas: con orientado para conexão e sem conexão. Nesta seção, veremos esses dois tipos e examinar as diferenças entre eles.

O serviço **orientado a conexão** é modelado após o sistema telefônico. Falar para alguém, você pega o telefone, disca o número, fala e depois desliga. Similarmente, para usar um serviço de rede orientado para conexão, o usuário do serviço primeiro estabelece termina uma conexão, usa a conexão e, em seguida, libera a conexão. o

aspecto essencial de uma conexão é que ela atua como um tubo: o remetente empurra os objetos (bits) em uma extremidade, e o receptor os retira na outra extremidade. Na maioria dos casos, a ordem é preservada para que os bits cheguem na ordem em que foram enviados. Em alguns casos, quando uma conexão é estabelecida, o remetente, o destinatário e o subnet conduzir uma **negociação** sobre os parâmetros a serem usados, como máximo tamanho da mensagem, qualidade do serviço exigido e outras questões. Normalmente, um lado faz uma proposta e o outro lado pode aceitá-la, rejeitá-la ou fazer uma contra proposta. Um **círculo** é outro nome para uma conexão com recursos associados, como uma largura de banda fixa. Isso data da rede telefônica em que um circuito era um caminho sobre um fio de cobre que conduzia uma conversa telefônica. Em contraste com o serviço orientado à conexão, o serviço sem **conexão** é modelado após o sistema postal. Cada mensagem (carta) carrega o endereço de destino completo,

Página 60

36

INTRODUÇÃO INDIVÍDUO. 1

e cada um é roteado através dos nós intermediários dentro do sistema independentemente de todas as mensagens subsequentes. Existem diferentes nomes para mensagens em contextos diferentes; um **pacote** é uma mensagem na camada de rede. Quando os nós mediadores recebem uma mensagem completa antes de enviá-la para o próximo nó, este é chamado **de comutação armazenar e encaminhar**. A alternativa, em que a transmissão de uma mensagem em um nó começa antes de ser completamente recebida pelo nó, é chamado **de comutação cut-through**. Normalmente, quando duas mensagens são enviadas para o mesmo destino, o primeiro enviado será o primeiro a chegar. No entanto, é possível que o primeiro enviado seja atrasado para que chegue o segundo.

Cada tipo de serviço pode ainda ser caracterizado por sua confiabilidade. Alguns serviços são confiáveis no sentido de que nunca perdem dados. Normalmente, um serviço confiável é implementado fazendo com que o receptor confirme o recebimento de cada mensagem para que o remetente tenha certeza de que ele chegou. O processo de reconhecimento apresenta sobrecarga e atrasos, que muitas vezes valem a pena, mas às vezes são indesejáveis.

Uma situação típica em que um serviço orientado à conexão confiável é apropriado é a transferência de arquivos. O proprietário do arquivo deseja ter certeza de que todos os bits chegarão

corretamente e na mesma ordem em que foram enviados. Muito poucos clientes de transferência de arquivos

preferiria um serviço que ocasionalmente embaralha ou perde alguns bits, mesmo que seja muito mais rápido.

O serviço orientado à conexão confiável tem duas variações menores: mensagem sequências e fluxos de bytes. Na primeira variante, os limites da mensagem são pré-servido. Quando duas mensagens de 1024 bytes são enviadas, elas chegam como duas mensagens distintas de 1024-

mensagens de byte, nunca como uma mensagem de 2048 bytes. No último, a conexão é simplesmente um fluxo de bytes, sem limites de mensagem. Quando 2.048 bytes chegam ao destinatário, não há como saber se foram enviadas como uma mensagem de 2048 bytes, duas mensagens de 1.024 bytes ou 2.048 mensagens de 1 byte. Se as páginas de um livro forem enviadas

em uma rede para um fotocompositor como mensagens separadas, pode ser importante preservar os limites da mensagem. Por outro lado, para baixar um filme em DVD, um fluxo de bytes do servidor para o computador do usuário é tudo o que é necessário. Mesmo limites sábios dentro do filme não são relevantes.

Para alguns aplicativos, os atrasos de trânsito introduzidos por confirmações são inaceitável. Uma dessas aplicações é o tráfego de voz digitalizado para **voz sobre IP**. Isto é menos perturbador para os usuários de telefone ouvirem um pouco de ruído na linha de vez em quando

tempo do que experimentar um atraso na espera por confirmações. Da mesma forma, quando transmitir uma videoconferência, ter alguns pixels errados não é problema, mas ter a imagem oscilando conforme o fluxo para e começa a corrigir os erros é irritante.

Nem todos os aplicativos requerem conexões. Por exemplo, spammers enviam eletronic junk-mail para muitos destinatários. O spammer provavelmente não quer ir ao trabalho de configurar e, posteriormente, destruir uma conexão com um destinatário apenas para enviar um item a eles. Nem é uma entrega 100 por cento confiável essencial, especialmente se custa mais. Tudo o que é necessário é uma forma de enviar uma única mensagem que tenha uma alta

Página 61

SEC. 1,3
SOFTWARE DE REDE

37

probabilidade de chegada, mas sem garantia. Não confiável (ou seja, não reconhecido) serviço sem conexão é frequentemente chamado de serviço de **datagrama**, em analogia com o telegrama

serviço, que também não retorna uma confirmação ao remetente. Apesar disso sendo não confiável, é a forma dominante na maioria das redes por razões que ficar claro mais tarde

Em outras situações, a conveniência de não ter que estabelecer uma conexão com enviar uma mensagem é desejado, mas a confiabilidade é essencial. O **reconhecido** serviço de **datagrama** pode ser fornecido para esses aplicativos. É como enviar um registro carta filtrada e solicitando aviso de recebimento. Quando o recibo volta, o remetente está absolutamente certo de que a carta foi entregue à parte pretendida e não perdido ao longo do caminho. Mensagens de texto em telefones celulares são um exemplo. Ainda outro serviço é o serviço de **solicitação-resposta**. Neste serviço o remetente transmite um único datagrama contendo uma solicitação; a resposta contém a resposta. A solicitação-resposta é comumente usada para implementar a comunicação no cliente-servidor modelo: o cliente emite uma solicitação e o servidor responde a ela. Por exemplo, um cliente de telefone móvel pode enviar uma consulta a um servidor de mapas para recuperar os dados do mapa

para a localização atual. A Figura 1-16 resume os tipos de serviços discutidos acima.

Fluxo de mensagens confiável
Fluxo de bytes confiável
Conexão não confiável
Datagrama não confiável
Datagrama reconhecido
Pedido-resposta
Serviço
Conexão-orientado
Conexão-Menos
Seqüência de páginas
Download de filme
Voz sobre IP
Lixo eletrônico
Mensagem de texto
Consulta de banco de dados

Exemplo

Figura 1-16. Seis diferentes tipos de serviço.

O conceito de usar comunicação não confiável pode ser confuso no início. Afinal, por que alguém preferiria uma comunicação não confiável a confiável comunicação? Em primeiro lugar, comunicação confiável (em nosso sentido, isto é, reconhecido) pode não estar disponível em uma determinada camada. Por exemplo, Ethernet faz não fornecem comunicação confiável. Os pacotes podem ocasionalmente ser danificados em transito. Cabe aos níveis de protocolo mais altos se recuperar desse problema. Em particular lar, muitos serviços confiáveis são construídos em cima de um serviço de datagrama não confiável. Sec- segundo, os atrasos inerentes ao fornecimento de um serviço confiável podem ser inaceitáveis, especialmente

especialmente em aplicativos de tempo real, como multimídia. Por essas razões, ambos relichocexistem comunicações capazes e não confiáveis.

Página 62

38

INTRODUÇÃO
INDIVÍDUO. 1

1.3.4 Primitivas de serviço

Um serviço é formalmente especificado por um conjunto de **primitivas** (operações) disponíveis para processos do usuário para acessar o serviço. Esses primitivos dizem ao serviço para executar alguma ação ou relatório sobre uma ação realizada por uma entidade par. Se a pilha de protocolo for localizados no sistema operacional, como costuma acontecer, os primitivos são normalmente do sistema

chamadas. Essas chamadas causam uma interceptação no modo kernel, que então transforma o controle do ma-

chine ao sistema operacional para enviar os pacotes necessários.

O conjunto de primitivas disponíveis depende da natureza do serviço que está sendo fornecido. Os primitivos para serviço orientado a conexão são diferentes daqueles de serviço sem conexão. Como um exemplo mínimo dos primitivos de serviço que

pode fornecer um fluxo de bytes confiável, considere as primitivas listadas na Figura 1.17.

Eles serão familiares aos fãs da interface de soquete de Berkeley, pois os primitivos são uma versão simplificada dessa interface.

Primitivo

Significado

OUÇO

Bloco à espera de uma conexão de entrada

CONECTAR

Estabeleça uma conexão com um colega em espera

ACEITAR

Aceite uma conexão de entrada de um par

RECEBER

Bloco à espera de uma mensagem recebida

ENVIAR

Envie uma mensagem para o colega

DESCONECTAR

Encerrar uma conexão

Figura 1-17. Seis primitivas de serviço que fornecem um simples orientado a conexão serviço.

Esses primitivos podem ser usados para uma interação de solicitação-resposta em um cliente-serviço. Para ilustrar como, esboçamos um protocolo simples que implementa o serviço usando datagramas reconhecidos.

Primeiro, o servidor executa LISTEN para indicar que está preparado para aceitar próximas conexões. Uma maneira comum de implementar LISTEN é torná-lo um bloco chamada de sistema. Depois de executar a primitiva, o processo do servidor é bloqueado até uma solicitação de conexão aparece.

Em seguida, o processo do cliente executa CONNECT para estabelecer uma conexão com o servidor. A chamada CONNECT precisa especificar a quem se conectar, então pode ter um parâmetro que fornece o endereço do servidor. O sistema operacional normalmente envia um pacote para o par, solicitando a conexão, conforme mostrado por (1) na Figura 1.18. O cliente o processo é suspenso até que haja uma resposta.

Quando o pacote chega ao servidor, o sistema operacional vê que o pacote et está solicitando uma conexão. Ele verifica se há um ouvinte e, em caso afirmativo, desbloqueia o ouvinte. O processo do servidor pode então estabelecer a conexão com a chamada ACCEPT . Isso envia uma resposta (2) de volta ao processo do cliente para aceitar o

Página 63

SEC. 1,3
SOFTWARE DE REDE

39

Máquina cliente

(1) Solicitação de conexão

(2) Aceitar resposta

Sistema

chamadas

Núcleo

Operativo

sistema

Cliente

processo

Motoristas

Protocolo

pilha

Máquina servidor

Sistema

processo

Núcleo

Motoristas

Protocolo

pilha

(3) Solicitação de dados

(4) Responder

(5) Desconectar

(6) Desconectar

Figura 1-18. Uma interação cliente-servidor simples usando datagramas reconhecidos.

conexão. A chegada dessa resposta libera o cliente. Neste ponto, o cliente e o servidor estão em execução e estabeleceram uma conexão.

A analogia óbvia entre este protocolo e a vida real é um cliente (cliente)

ligando para o gerente de atendimento ao cliente de uma empresa. No início do dia, o serviço gerente se senta ao lado de seu telefone para o caso de ele tocar. Mais tarde, um cliente faz uma chamada.

Quando o gerente pega o telefone, a conexão é estabelecida.

A próxima etapa é para o servidor executar RECEIVE para se preparar para aceitar o primeiro solicitação. Normalmente, o servidor faz isso imediatamente após ser liberado do LISTEN , antes que a confirmação possa voltar ao cliente. A chamada RECEIVE bloqueia o servidor.

Em seguida, o cliente executa SEND para transmitir sua solicitação (3) seguido pelo ex-execução de RECEIVE para obter a resposta. A chegada do pacote de solicitação no servidor máquina desbloqueia o servidor para que possa lidar com a solicitação. Depois de ter feito o funcionar, o servidor usa SEND para devolver a resposta ao cliente (4). A chegada de este pacote desbloqueia o cliente, que agora pode inspecionar a resposta. Se o cliente tem pedidos adicionais, pode fazê-los agora.

Quando o cliente termina , ele executa DISCONNECT para encerrar a conexão

(5). Normalmente, um DISCONNECT inicial é uma chamada de bloqueio, suspendendo o cliente e enviar um pacote ao servidor informando que a conexão não é mais necessária.

Quando o servidor recebe o pacote, ele também emite um DISCONNECT próprio, ack-agudizar o cliente e liberar a conexão (6). Quando o pacote do servidor volta para a máquina cliente, o processo cliente é liberado e a conexão é partido. Em suma, é assim que funciona a comunicação orientada a conexão.

Claro, a vida não é tão simples. Muitas coisas podem dar errado aqui. A temporização pode estar errado (por exemplo, o CONNECT é feito antes do LISTEN), os pacotes podem ser perdidos, e muito mais. Veremos esses problemas em grande detalhe posteriormente, mas para o momento, a Figura 1-18 resume brevemente como a comunicação cliente-servidor pode trabalhar com datagramas reconhecidos para que possamos ignorar pacotes perdidos.

Dado que seis pacotes são necessários para completar este protocolo, pode-se perguntar por que um protocolo sem conexão não é usado em vez disso. A resposta é que em um mundo perfeito que pudesse ser, caso em que apenas dois pacotes seriam necessários: um

ausência de? Como o cliente saberia se o último pacote realmente recebeu foi realmente o último pacote enviado? Suponha que o cliente deseje um segundo arquivo. Como poderia distinguir o pacote 1 do segundo arquivo de um pacote perdido 1 do primeiro arquivo que de repente encontrou seu caminho para o cliente? Em suma, no mundo real, uma simples revisão O protocolo de resposta de busca em uma rede não confiável é freqüentemente inadequado. No cap. 3

vamos estudar uma variedade de protocolos em detalhes que superam esses e outros problemas lems. Por enquanto, basta dizer que ter um fluxo de bytes confiável e ordenado entre processos às vezes é muito conveniente.

1.3.5 A relação dos serviços com os protocolos

Serviços e protocolos são conceitos distintos. Esta distinção é tão importante que o enfatizamos novamente aqui. Um *serviço* é um conjunto de primitivas (operações) que uma camada fornece à camada acima dela. O serviço define quais operações o camada está preparada para atuar em nome de seus usuários, mas não diz nada sobre como essas operações são implementadas. Um serviço está relacionado a uma interface entre duas camadas, com a camada inferior sendo o provedor de serviços e a camada superior ser o usuário do serviço.

Um *protocolo*, em contraste, é um conjunto de regras que regem o formato e o significado de os pacotes ou mensagens que são trocados pelas entidades pares dentro de uma camada.

As entidades usam protocolos para implementar suas definições de serviço. Eles são livres para alterar seus protocolos à vontade, desde que não altere o serviço visível para seus usuários. Desta forma, o serviço e o protocolo são completamente dissociados.

Este é um conceito chave que qualquer designer de rede deve entender bem.

Para repetir este ponto crucial, os serviços se relacionam às interfaces entre as camadas, como ilustrado na Figura 1-19. Em contraste, os protocolos se relacionam aos pacotes enviados entre entidades de mesmo nível em máquinas diferentes. É muito importante não confundir os dois conceitos.

Vale a pena fazer uma analogia com as linguagens de programação. Um serviço é como um tipo de dado abstrato ou um objeto em uma linguagem orientada a objetos. Ele define a operações que podem ser realizadas em um objeto, mas não especifica como essas operações são implementados. Em contraste, um protocolo está relacionado à *implementação* do serviço e, como tal, não é visível para o usuário do serviço.

Muitos protocolos mais antigos não distinguiam o serviço do protocolo. Em ef De fato, uma camada típica pode ter tido um serviço primitivo SEND PACKET com o usuário fornecer um ponteiro para um pacote totalmente montado. Este arranjo significava que todos as mudanças no protocolo eram imediatamente visíveis aos usuários. A maioria das redes de os signatários agora consideram tal projeto um erro grave.

SEC. 1,4
MODELOS DE REFERÊNCIA

41

Camada k
Camada k + 1
Camada k - 1
Protocolo
Serviço prestado pela camada k
Camada k
Camada k + 1
Camada k - 1

Figura 1-19. A relação entre um serviço e um protocolo.

1.4 MODELOS DE REFERÊNCIA

Agora que discutimos as redes em camadas de forma abstrata, é hora de olhar em alguns exemplos. Discutiremos duas arquiteturas de rede importantes: o OSI modelo de referência e o modelo de referência TCP / IP. Embora os *protocolos* associados com o modelo OSI não são mais usados, o *modelo em si* é bastante geral e ainda válido, e os recursos discutidos em cada camada ainda são muito importantes. O modelo TCP / IP tem as propriedades opostas: o modelo em si não é de muito útil, mas os protocolos são amplamente utilizados. Por esta razão, vamos olhar para ambos

deles em detalhes. Além disso, às vezes você pode aprender mais com as falhas do que com sucessos.

1.4.1 O Modelo de Referência OSI

O modelo OSI (sem o meio físico) é mostrado na Figura 1.20. Isto modelo é baseado em uma proposta desenvolvida pela International Standards Organization (ISO) como um primeiro passo para a padronização internacional dos protocolos usados nas várias camadas (Day e Zimmermann, 1983). Foi revisado em 1995 (dia, 1995). O modelo é denominado ISO **OSI (Open Systems Interconnection) Reference** o modelo porque lida com a conexão de sistemas abertos, ou seja, sistemas que estão abertos para comunicação com outros sistemas. Vamos chamá-lo de **OSI modelo** para breve.

O modelo OSI possui sete camadas. Os princípios que foram aplicados para chegar a as sete camadas podem ser resumidas da seguinte forma:

1. Uma camada deve ser criada onde uma abstração diferente é necessária.
2. Cada camada deve executar uma função bem definida.
3. A função de cada camada deve ser escolhida tendo em vista definição de protocolos padronizados internacionalmente.

Página 66

42

INTRODUÇÃO

INDIVÍDUO. 1

Camada

Apresentação

Inscrição

Sessão

Transporte

Rede

Link de dados

Física

7

6

5

4

3

2

1

Interface

Host A

Nome da unidade

trocado

APDU

PPDU

SPDU

TPDU

Pacote

Quadro, Armação

Mordeu

Apresentação

Inscrição

Sessão

Transporte

Rede

Link de dados

Física

Host B

Rede

Rede

Link de dados

Link de dados

Física

Física

Roteador

Roteador

Protocolo de sub-rede interna

Protocolo de aplicação

Protocolo de apresentação

Protocolo de transporte

Protocolo de sessão

Límite da sub-rede de comunicação

Protocolo host-roteador da camada de rede

Protocolo host-roteador da camada de enlace

Protocolo host-roteador da camada física

Figura 1-20. O modelo de referência OSI.

4. Os limites da camada devem ser escolhidos para minimizar as informações fluir pelas interfaces.

5. O número de camadas deve ser grande o suficiente para funções distintas não precisam ser jogados juntos na mesma camada por necessidade e pequeno o suficiente para que a arquitetura não se torne pesada.

Abaixo, discutiremos cada camada do modelo por vez, começando na parte inferior camada. Observe que o modelo OSI em si não é uma arquitetura de rede porque não especifica os serviços e protocolos exatos a serem usados em cada camada. Apenas diz o que cada camada deve fazer. No entanto, a ISO também produziu padrões para todos os camadas, embora não façam parte do próprio modelo de referência. Cada um tem publicado como um padrão internacional separado. O *modelo* (em parte) é amplamente usado, embora os protocolos associados tenham sido esquecidos.

Página 67

SEC. 1,4
MODELOS DE REFERÊNCIA

43

A Camada Física

A **camada física** está preocupada em transmitir bits brutos através de uma comunicação canal de cátions. Os problemas de design têm a ver com garantir que, quando um lado envia um bit 1, ele é recebido pelo outro lado como um bit 1, não como um bit 0. Perguntas típicas aqui são quais sinais elétricos devem ser usados para representar 1 e 0, como muitos nanosegundos dura um bit, se a transmissão pode prosseguir simultaneamente em ambas as direções, como a conexão inicial é estabelecida, como é interrompida quando ambos os lados estiverem prontos, quantos pinos o conector de rede tem e quais cada pino é usado. Essas questões de design lidam principalmente com mecânica, elétrica, e interfaces de temporização, bem como o meio de transmissão físico, que se encontra abaixo da camada física.

A camada de link de dados

A principal tarefa da **camada de enlace de dados** é transformar uma instalação de transmissão bruta em uma linha que parece livre de erros de transmissão não detectados. Ele faz isso por mascarando os erros reais para que a camada de rede não os veja. Realiza esta tarefa fazendo com que o remetente divida os dados de entrada em **quadros de dados** (normalmente

algumas centenas ou alguns milhares de bytes) e transmitem os quadros sequencialmente. E se o serviço é confiável, o receptor confirma o recebimento correto de cada quadro por meio de envio-retornando um **quadro de confirmação**.

Outro problema que surge na camada de enlace de dados (e na maioria das camadas superiores também) é como evitar que um transmissor rápido afogue um receptor lento em dados.

Pode ser necessário algum mecanismo de regulação de tráfego para permitir que o transmissor saiba quando o receptor pode aceitar mais dados.

As redes de transmissão têm um problema adicional na camada de enlace de dados: como controlar o acesso ao canal compartilhado. Uma subcamada especial da camada de enlace de dados, o

subcamada de **controle de acesso ao meio**, lida com este problema.

A Camada de Rede

A **camada de rede** controla a operação da sub-rede. Uma questão chave de design é determinar como os pacotes são roteados da origem ao destino. As rotas podem ser com base em tabelas estáticas que são "conectadas" à rede e raramente alteradas, ou com mais frequência, eles podem ser atualizados automaticamente para evitar componentes com falha. Eles

também pode ser determinado no início de cada conversa, por exemplo, um terminal sessão, como um login em uma máquina remota. Finalmente, eles podem ser altamente dinâmicos, sendo determinado novamente para cada pacote para refletir a carga de rede atual.

Se muitos pacotes estiverem presentes na sub-rede ao mesmo tempo, eles entrarão uns dos outros, formando gargalos. Lidar com o congestionamento também é uma responsabilidade da camada de rede, em conjunto com camadas superiores que adaptam a carga

44

INTRODUÇÃO INDIVÍDUO. 1

eles colocam na rede. De forma mais geral, a qualidade do serviço prestado (atraso, tempo de trânsito, jitter, etc.) também é um problema da camada de rede. Quando um pacote precisa viajar de uma rede para outra para chegar ao seu destino, muitos problemas podem surgir. O endereçamento usado pela segunda rede pode ser diferente daquele usado pelo primeiro. O segundo pode não aceitar o pacote porque é muito grande. Os protocolos podem ser diferentes e assim por diante. Está acima para a camada de rede para superar todos esses problemas para permitir uma rede heterogênea funciona para ser interconectado.

Em redes de broadcast, o problema de roteamento é simples, então a camada de rede é muitas vezes fino ou mesmo inexistente.

A Camada de Transporte

A função básica da **camada de transporte** é aceitar dados de cima dela, dividir em unidades menores, se necessário, passe-as para a camada de rede e garanta que todas as peças chegam corretamente na outra extremidade. Além disso, tudo isso deve ser feito de forma eficiente e de uma forma que isole as camadas superiores das mudanças inevitáveis na tecnologia de hardware ao longo do tempo.

A camada de transporte também determina que tipo de serviço deve ser prestado à equipe camada de ação e, em última análise, para os usuários da rede. O tipo mais popular de conexão de transporte é um canal ponto a ponto sem erros que entrega mensagens ou bytes na ordem em que foram enviados. No entanto, outros tipos possíveis de serviço de transporte existe, como o transporte de mensagens isoladas sem garantia sobre a ordem de entrega e a transmissão de mensagens para vários destinos. O tipo de serviço é determinado quando a conexão é established. (À parte, um canal livre de erros é completamente impossível de alcançar; o que as pessoas realmente querem dizer com este termo é que a taxa de erro é baixa o suficiente para ignorar na prática.)

A camada de transporte é uma verdadeira camada ponta a ponta; ele carrega dados desde da origem ao destino. Em outras palavras, um programa na máquina de origem mantém uma conversa com um programa semelhante na máquina de destino, usando os cabeçalhos das mensagens e as mensagens de controle. Nas camadas inferiores, cada protocolo é entre uma máquina e seus vizinhos imediatos, e não entre o último máquinas de origem e de destino, que podem ser separadas por muitos roteadores. A diferença entre as camadas 1 a 3, que são encadeadas, e as camadas 4 a 7, que são de ponta a ponta, é ilustrado na Fig. 1-20.

A Camada de Sessão

A camada de sessão permite que usuários em máquinas diferentes estabeleçam **sessões** antes de entre eles. As sessões oferecem vários serviços, incluindo **controle de diálogo** (mantendo controle de quem é a vez de transmitir), **gerenciamento de tokens** (evitando duas partes de tentar a mesma operação crítica simultaneamente) e **sincronização**

SEC. 1,4

MODELOS DE REFERÊNCIA

45

(marcar transmissões longas para permitir que continuem de onde saíram desligado em caso de falha e recuperação subsequente).

A Camada de Apresentação

Ao contrário das camadas inferiores, que se preocupam principalmente em mover os bits, a **camada de apresentação** está preocupada com a sintaxe e semântica da informação informação transmitida. A fim de tornar possível para computadores com diferentes in-

representações de dados ternal para se comunicar, as estruturas de dados a serem trocadas pode ser definido de forma abstrata, juntamente com uma codificação padrão a ser usada " em o fio. " A camada de apresentação gerencia essas estruturas de dados abstratos e al- permite que estruturas de dados de nível superior (por exemplo, registros bancários) sejam definidas e trocado.

A Camada de Aplicação

A **camada de aplicação** contém uma variedade de protocolos que são comumente necessários para os usuários. Um protocolo de aplicação amplamente utilizado é o **HTTP (HyperText Protocol de Transferência)**, que é a base da World Wide Web. Quando um

navegador deseja uma página da Web, ele envia o nome da página que deseja para o servidor hospedar a página usando HTTP. O servidor então envia a página de volta. Outros aplicativos os protocolos de comunicação são usados para transferência de arquivos, correio eletrônico e notícias de rede.

1.4.2 O modelo de referência TCP / IP

Vamos agora passar do modelo de referência OSI para o modelo de referência usado em o avô de todas as redes de computadores de longa distância, a ARPANET, e seu sucessor, a Internet mundial. Embora possamos dar uma breve história do ARPANET mais tarde, é útil mencionar alguns aspectos-chave agora. o ARPANET era uma rede de pesquisa patrocinada pelo DoD (Departamento de Defesa). Acabou conectando centenas de universidades e instituições governamentais instalações, usando linhas de telefone alugadas. Quando as redes de satélite e rádio eram adicionado posteriormente, os protocolos existentes tiveram problemas para interagir com eles, então um novo

arquitetura de referência era necessária. Assim, quase desde o início, a capacidade de conectar várias redes de maneira contínua era um dos principais objetivos do projeto.

Essa arquitetura mais tarde ficou conhecida como **Modelo de Referência TCP / IP**, após sua dois protocolos primários. Foi descrito pela primeira vez por Cerf e Kahn (1974), e mais tarde refinado e definido como um padrão na comunidade da Internet (Braden, 1989). o

A filosofia de design por trás do modelo é discutida por Clark (1988).

Dada a preocupação do DoD de que alguns de seus preciosos hosts, roteadores e internet-gateways de trabalho podem explodir em pedaços a qualquer momento por um ataque de a União Soviética, outro objetivo importante era que a rede pudesse sobreviver à perda de hardware de sub-rede, sem que as conversas existentes sejam interrompidas. Em outro

46

INTRODUÇÃO

INDIVÍDUO. 1

palavras, o DoD queria que as conexões permanecessem intactas enquanto a fonte e máquinas de destino estavam funcionando, mesmo se algumas das máquinas ou transmissões as linhas intermediárias foram repentinamente desativadas. Além disso, uma vez que aplicações com requisitos divergentes foram previstas, variando de transferência arquivos para transmissão de voz em tempo real, uma arquitetura flexível era necessária.

A Camada de Link

Todos esses requisitos levaram à escolha de uma rede de comutação de pacotes baseada em uma camada sem conexão que funciona em redes diferentes. A camada mais baixa em o modelo, a **camada de link** descreve quais links, como linhas seriais e clássicos Ethernet. A Internet deve fazer para atender às necessidades dessa camada de internet sem conexão. Não é realmente uma camada, no sentido normal do termo, mas sim uma interface entre hosts e links de transmissão. O material inicial sobre o modelo TCP / IP tem pouco para dizer sobre isso.

A camada da Internet

A **camada da Internet** é o eixo que mantém toda a arquitetura unida.

Ele é mostrado na Figura 1.21 como correspondendo aproximadamente à camada de rede OSI. Está trabalhando permitir que hosts injetem pacotes em qualquer rede e os façam viajar em

dependendo do destino (potencialmente em uma rede diferente). Eles podem chegam em uma ordem completamente diferente daquela em que foram enviados, caso em que é o trabalho das camadas superiores para reorganizá-los, se a entrega em ordem for desejada. Nota que "internet" é usado aqui em um sentido genérico, embora esta camada esteja presente em a Internet.

TCP / IP	
OSI	
Inscrição	
Apresentação	
Sessão	
Transporte	
Rede	
Link de dados	
Física	
7	
6	
5	
4	
3	
2	
1	
Inscrição	
Transporte	
Internet	
Ligação	
Não presente	
no modelo	

Figura 1-21. O modelo de referência TCP / IP.

A analogia aqui é com o sistema de correio (tradicional). Uma pessoa pode deixar cair um sequência de cartas internacionais em uma caixa de correio em um país, e com um pouco de sorte,

Página 71

SEC. 1,4

MODELOS DE REFERÊNCIA

47

a maioria deles será entregue no endereço correto do país de destino.

As cartas provavelmente passarão por um ou mais gateways de correio internacionais ao longo do caminho, mas isso é transparente para os usuários. Além disso, que cada país (ou seja, cada rede) tem seus próprios selos, tamanhos de envelope preferidos e entrega as regras são ocultadas dos usuários.

A camada de internet define um formato de pacote oficial e protocolo chamado **IP** (**Internet Protocol**), mais um protocolo complementar denominado **ICMP** (**Internet Control Protocol de Mensagem**) que o ajuda a funcionar. O trabalho da camada da Internet é entregar pacotes IP para onde eles deveriam ir. O roteamento de pacotes é claramente um problema principal aqui, assim como o congestionamento (embora o IP não tenha se mostrado eficaz em evitar congestionamento).

A Camada de Transporte

A camada acima da camada de internet no modelo TCP / IP agora é normalmente chamada a **camada de transporte**. Ele é projetado para permitir entidades pares na origem e no destino hosts nacionais para manter uma conversa, assim como na camada de transporte OSI. Dois protocolos de transporte ponta a ponta foram definidos aqui. O primeiro, **TCP**

(**Transmission Control Protocol**), é um protocolo orientado a conexão confiável que permite que um fluxo de bytes originado em uma máquina seja entregue sem erros em qualquer outra máquina na Internet. Ele segmenta o fluxo de bytes de entrada em mensagens discretas e passa cada uma para a camada de internet. No destino, o processo TCP receptor reagrupa as mensagens recebidas na saída corrente. O TCP também lida com o controle de fluxo para garantir que um remetente rápido não atrapalhe um

receptor lento com mais mensagens do que pode controlar.

O segundo protocolo nesta camada, **UDP** (**User Datagram Protocol**), é um protocolo não confiável e sem conexão para aplicativos que não desejam TCP sequenciamento ou controle de fluxo e desejam fornecer os seus próprios. Também é amplamente utilizado

para consultas e aplicativos únicos do tipo cliente-servidor-resposta em que entrega rápida é mais importante do que entrega precisa, como transmitir discurso ou vídeo. A relação de IP, TCP e UDP é mostrada na Figura 1.22. Desde a o modelo foi desenvolvido, o IP foi implementado em muitas outras redes.

A Camada de Aplicação

O modelo TCP / IP não possui camadas de sessão ou apresentação. Não há necessidade de eles foram percebidos. Em vez disso, os aplicativos simplesmente incluem qualquer sessão e pres-funções de entação que eles requerem. A experiência com o modelo OSI provou esta visão correta: essas camadas são de pouca utilidade para a maioria dos aplicativos. No topo da camada de transporte está a **camada de aplicação**. Ele contém todos os protocolos de nível superior. Os primeiros incluíam terminal virtual (TELNET), arquivo transfer (FTP) e correio eletrônico (SMTP). Muitos outros protocolos foram adicionados ao ao longo dos anos. Alguns importantes que estudaremos, mostrados na Fig. 1-22,

Página 72

48

INTRODUÇÃO

INDIVÍDUO. 1

Ligaçao

Ethernet

802.11

SONET

DSL

IP

ICMP

HTTP

RTP

SMTP

DNS

TCP

UDP

Internet

Transporte

Camadas

Protocolos

Inscrição

Figura 1-22. O modelo TCP / IP com alguns protocolos que estudaremos.

incluem o Domain Name System (DNS), para mapear nomes de host em sua rede endereços de trabalho, HTTP, o protocolo de busca de páginas na World Wide Web, e RTP, o protocolo para entrega de mídia em tempo real, como voz ou filmes.

1.4.3 O modelo usado neste livro

Como mencionado anteriormente, a força do modelo de referência OSI é o *modelo ele-self* (sem as camadas de apresentação e sessão), que provou ser excepcionalmente útil para discutir redes de computadores. Em contraste, a força de o modelo de referência TCP / IP são os *protocolos*, que têm sido amplamente usados para muitos anos. Como os cientistas da computação gostam de ter seu bolo e comê-lo também, nós usará o modelo híbrido da Figura 1.23 como estrutura para este livro.

5

Inscrição

4

Transporte

3

Rede

2

Ligaçao

1

Física

Figura 1-23. O modelo de referência usado neste livro.

Este modelo tem cinco camadas, indo da camada física até o camadas de link, rede e transporte para a camada de aplicação. A camada física especifica como transmitir bits em diferentes tipos de mídia como elétrica (ou outros sinais analógicos). A camada de link está preocupada em como enviar comprimento finito mensagens entre computadores conectados diretamente com níveis específicos de confiabilidade ity. Ethernet e 802.11 são exemplos de protocolos de camada de link.

SEC. 1,4
MODELOS DE REFERÊNCIA

49

A camada de rede trata de como combinar vários links em redes, e redes de redes, em internetworks para que possamos enviar pacotes entre computadores distantes. Isso inclui a tarefa de encontrar o caminho ao longo do qual enviar os pacotes. IP é o protocolo de exemplo principal que estudaremos para esta camada. A camada de transporte fortalece as garantias de entrega da camada de rede, geralmente com maior confiabilidade e fornecer abstrações de entrega, como um confiável fluxo de bytes, que atende às necessidades de diferentes aplicativos. TCP é um importante exemplo de um protocolo de camada de transporte.

Por fim, a camada de aplicativo contém programas que fazem uso da rede.

Muitos, mas não todos, os aplicativos em rede têm interfaces de usuário, como um Web navegador. Nossa preocupação, no entanto, é com a parte do programa que usa a rede. Este é o protocolo HTTP no caso do navegador da web. Tem também importantes programas de suporte na camada de aplicação, como o DNS, que são usados por muitos aplicativos.

Nossa sequência de capítulos é baseada neste modelo. Desta forma, retemos o valor do modelo OSI para entender arquiteturas de rede, mas concentre-se antes marily em protocolos que são importantes na prática, de TCP / IP e protocols para os mais recentes, como 802.11, SONET e Bluetooth.

1.4.4 Uma comparação dos modelos de referência OSI e TCP / IP

Os modelos de referência OSI e TCP / IP têm muito em comum. Ambos são baseado no conceito de uma pilha de protocolos independentes. Além disso, a funcionalidade das camadas é aproximadamente semelhante. Por exemplo, em ambos os modelos as camadas acima através e incluindo a camada de transporte existem para fornecer uma rede de ponta a ponta serviço de transporte independente do trabalho para processos que desejam se comunicar. Estes camadas formam o provedor de transporte. Novamente em ambos os modelos, as camadas acima do trans-

desporto são utilizadores orientados para a aplicação do serviço de transporte.

Apesar dessas semelhanças fundamentais, os dois modelos também têm muitas diferenças ferências. Nesta seção, vamos nos concentrar nas principais diferenças entre os dois referentes modelos de referência. É importante observar que estamos comparando os *modelos de referência* aqui, não as *pilhas de protocolo* correspondentes . Os próprios protocolos serão discutido mais tarde. Para um livro inteiro comparando e contrastando TCP / IP e OSI, consulte Piscitello e Chapin (1993).

Três conceitos são centrais para o modelo OSI:

1. Serviços.
2. Interfaces.
3. Protocolos.

Provavelmente, a maior contribuição do modelo OSI é que ele faz a distinção entre esses três conceitos explícitos. Cada camada executa alguns serviços para o

50

INTRODUÇÃO
INDIVÍDUO. 1

camada acima dela. A definição do serviço diz o que a camada faz, não como as entidades acima dele acesse ou como a camada funciona. Ele define a semântica da camada.

A interface de uma camada informa aos processos acima dela como acessá-la. Especifica quais são os parâmetros e quais resultados esperar. Também não diz nada sobre como a camada funciona por dentro.

Finalmente, os protocolos de mesmo nível usados em uma camada são da própria responsabilidade da camada. Pode

usar qualquer protocolo que desejar, desde que faça o trabalho (ou seja, fornece o

serviços oferecidos). Ele também pode alterá-los à vontade, sem afetar o software em camadas superiores.

Essas idéias se encaixam muito bem com as idéias modernas sobre programas orientados a objetos programming. Um objeto, como uma camada, tem um conjunto de métodos (operações) que processos fora do objeto podem invocar. A semântica desses métodos define o conjunto de serviços que o objeto oferece. Os parâmetros e resultados dos métodos formam o interface do objeto. O código interno ao objeto é seu protocolo e não é visível ou de qualquer preocupação fora do objeto.

O modelo TCP / IP originalmente não distingua claramente entre os serviços, interfaces e protocolos, embora as pessoas tenham tentado retrofit após o fato para torná-lo mais parecido com o OSI. Por exemplo, os únicos serviços reais oferecidos pela internet camada são SEND IP PACKET e RECEIVE IP PACKET . Como consequência, o protocolos no modelo OSI são melhor escondidos do que no modelo TCP / IP e podem ser substituídos com relativa facilidade conforme a tecnologia muda. Ser capaz de fazer tal mudanças de forma transparente é um dos principais objetivos de ter protocolos em camadas em o primeiro lugar.

O modelo de referência OSI foi desenvolvido *antes* dos protocolos correspondentes foram inventados. Essa ordenação significava que o modelo não era tendencioso para um conjunto particular de protocolos, um fato que o tornou bastante geral. A desvantagem disso pedido era que os designers não tinham muita experiência com o assunto e não tinha uma boa ideia de qual funcionalidade colocar em qual camada.

Por exemplo, a camada de enlace de dados originalmente lidava apenas com rede ponto a ponto trabalho. Quando as redes de transmissão surgiram, uma nova subcamada teve que ser adicionada no modelo. Além disso, quando as pessoas começaram a construir redes reais usando o Modelo OSI e protocolos existentes, descobriu-se que essas redes não corresponder às especificações de serviço necessárias (maravilha das maravilhas), portanto, a convergência

subcamadas tiveram que ser enxertadas no modelo para fornecer um lugar para cobrir as diferenças. Finalmente, o comitê originalmente esperava que cada país teria uma rede, administrada pelo governo e usando os protocolos OSI, então nenhum pensamento foi dado à internetworking. Para encurtar a história, as coisas não saíram dessa maneira.

Com o TCP / IP, o inverso era verdadeiro: os protocolos vinham primeiro, e o modelo era na verdade, apenas uma descrição dos protocolos existentes. Não houve nenhum problema com o protocolos adequados ao modelo. Eles se encaixam perfeitamente. O único problema era que o modelo não se encaixava em nenhuma outra pilha de protocolo. Consequentemente, não foi especialmente útil para descrever outras redes não TCP / IP.

Página 75

SEC. 1,4 MODELOS DE REFERÊNCIA

51

Passando de questões filosóficas para outras mais específicas, uma diferença óbvia diferença entre os dois modelos é o número de camadas: o modelo OSI tem sete camadas e o modelo TCP / IP tem quatro. Ambos têm rede (inter), transporte e camadas de aplicação, mas as outras camadas são diferentes.

Outra diferença está na área de sem conexão versus orientada a conexão comunicação. O modelo OSI oferece suporte tanto sem conexão quanto sem conexão comunicação orientada na camada de rede, mas apenas orientada a conexão com comunicação na camada de transporte, onde conta (porque o serviço de transporte é visível para os usuários). O modelo TCP / IP suporta apenas um modo na rede camada (sem conexão), mas ambos na camada de transporte, dando aos usuários uma escolha. Essa escolha é especialmente importante para protocolos simples de solicitação-resposta.

1.4.5 Uma crítica do modelo OSI e protocolos

Nem o modelo OSI e seus protocolos, nem o modelo TCP / IP e seus protocolos são perfeitos. Muitas críticas podem ser, e têm sido, dirigidas a ambos

eles. Nesta seção e na próxima, veremos algumas dessas críticas. Começaremos com o OSI e examinaremos o TCP / IP posteriormente. Na época em que a segunda edição deste livro foi publicada (1989), ele apareceu a muitos especialistas na área que o modelo OSI e seus protocolos dominar o mundo e tirar todo o resto do caminho. Isso não aconteceu caneta. Por quê? Uma revisão de alguns dos motivos pode ser útil. Eles podem ser somados marizado como:

1. Momento ruim.
2. Tecnologia ruim.
3. Implementações ruins.
4. Política ruim.

Mau momento

Primeiro, vejamos a razão um: o momento errado. O momento em que um padrão é estabelecido é absolutamente crítico para seu sucesso. David Clark do MIT tem uma teoria de padrões que ele chama de *apocalipse dos dois elefantes*, que é ilustrado na Figura 1-24.

Esta figura mostra a quantidade de atividade em torno de um novo assunto. Quando o assunto é descoberto pela primeira vez, há uma explosão de atividade de pesquisa na forma de discussões, documentos e reuniões. Depois de um tempo, essa atividade diminui, corporações descobrem o assunto e a onda de investimentos de bilhões de dólares.

É essencial que os padrões sejam escritos na calha entre os dois "elefantes". Se forem escritos muito cedo (antes de os resultados da pesquisa estarem bem

Página 76

52

INTRODUÇÃO

INDIVÍDUO. 1

Tempo

Atividade

Pesquisa

Padrões

Bilhões de dólares

investimento

Figura 1-24. O apocalipse dos dois elefantes.

estabelecido), o assunto ainda pode ser mal compreendido; o resultado é um mau padrão dard. Se forem escritos tarde demais, muitas empresas já podem ter feito ma- para investimentos em diferentes maneiras de fazer as coisas que os padrões sejam eficazes ignorado. Se o intervalo entre os dois elefantes for muito curto (porque todos está com pressa para começar), as pessoas que desenvolvem os padrões podem ser esmagadas.

Agora parece que os protocolos OSI padrão foram esmagados. O competidor Os protocolos TCP / IP já estavam em uso generalizado por universidades de pesquisa pelo vez que os protocolos OSI apareceram. Enquanto a onda de investimentos de bilhões de dólares ainda não atingido, o mercado acadêmico era grande o suficiente para que muitos fornecedores começaram

oferecendo produtos TCP / IP com cautela. Quando o OSI apareceu, eles não queriam para suportar uma segunda pilha de protocolo até que fossem forçados, então não havia ofertas iniciais. Com todas as empresas esperando que todas as outras vão primeiro, nenhuma empresa foi a primeira e o OSI nunca aconteceu.

Tecnologia Ruim

A segunda razão pela qual o OSI nunca pegou é que tanto o modelo quanto o os protocolos são falhos. A escolha de sete camadas foi mais política do que técnica cal, e duas das camadas (sessão e apresentação) estão quase vazias, enquanto dois outros (link de dados e rede) estão lotados.

O modelo OSI, junto com suas definições de serviço e protocolos associados, é extraordinariamente complexo. Quando empilhados, os padrões impressos ocupam uma significativa não posso fração de um metro de papel. Eles também são difíceis de implementar e ineficazes eficiente em operação. Neste contexto, um enigma colocado por Paul Mockapetris e citado por Rose (1993) vem à mente:

P: O que você ganha quando cruza um mafioso com um padrão internacional?

R: Alguém que lhe faz uma oferta que você não consegue entender.

Página 77

SEC. 1,4

MODELOS DE REFERÊNCIA

53

Além de ser incompreensível, outro problema com o OSI é que alguns funções, como endereçamento, controle de fluxo e controle de erros, reaparecem novamente e novamente em cada camada. Saltzer et al. (1984), por exemplo, apontaram que ser eficaz, o controle de erros deve ser feito na camada mais alta, de modo que repeti-lo e mais em cada uma das camadas inferiores é freqüentemente desnecessário e ineficiente.

Implementações Ruins

Dada a enorme complexidade do modelo e dos protocolos, virá como nenhuma surpresa que as implementações iniciais foram enormes, pesadas e lentas. Todos que os experimentaram se queimaram. Não demorou muito para as pessoas associarem "OSI" com "baixa qualidade." Embora os produtos tenham melhorado no decorrer de tempo, a imagem travou.

Em contraste, uma das primeiras implementações de TCP / IP foi parte de Berkeley UNIX e era muito bom (para não mencionar, gratuito). As pessoas começaram a usar rapidamente, o que levou a uma grande comunidade de usuários, o que levou a melhorias, o que levou a um comunidade ainda maior. Aqui, a espiral era para cima em vez de para baixo.

Política Ruim

Por conta da implantação inicial, muitas pessoas, principalmente na academia, pensava no TCP / IP como parte do UNIX e UNIX na década de 1980 na academia não era diferente da paternidade (então chamada incorretamente de maternidade) e da torta de maçã. OSI, por outro lado, era amplamente considerado uma criatura da Europa ministérios de telecomunicações, a Comunidade Europeia e, posteriormente, o governo dos EUA governament. Essa crença era apenas parcialmente verdadeira, mas a própria ideia de um bando de governantes

burocratas mentais tentando empurrar um padrão tecnicamente inferior goela abaixo dos pesquisadores e programadores pobres nas trincheiras realmente desenvolvem redes de computadores não ajudaram a causa da OSI. Algumas pessoas viram este desenvolvimento à mesma luz que a IBM anunciou na década de 1960 que a PL / I era a linguagem do futuro, ou o DoD corrigindo isso mais tarde, anunciando que era na verdade, Ada.

1.4.6 Uma crítica do modelo de referência TCP / IP

O modelo e os protocolos TCP / IP também têm seus problemas. Primeiro, o modelo não distingue claramente os conceitos de serviços, interfaces e protocolos.

Boas práticas de engenharia de software requerem a diferenciação entre as especificação e implementação, algo que o OSI faz com muito cuidado, mas TCP / IP não. Consequentemente, o modelo TCP / IP não é muito um guia para desassinatura de novas redes usando novas tecnologias.

Em segundo lugar, o modelo TCP / IP não é geral e é pouco adequado para descrever em qualquer pilha de protocolo diferente de TCP / IP. Tentando usar o modelo TCP / IP para descrever o Bluetooth, por exemplo, é completamente impossível.

Página 78

54

INTRODUÇÃO

INDIVÍDUO. 1

Terceiro, a camada de link não é realmente uma camada no sentido normal do termo conforme usado no contexto de protocolos em camadas. É uma interface (entre a rede e camadas de link de dados). A distinção entre uma interface e uma camada é crucial, e não se deve ser descuidado sobre isso.

Quarto, o modelo TCP / IP não faz distinção entre o físico e o de dados camadas de link. São completamente diferentes. A camada física tem a ver com o

características de transmissão de fio de cobre, fibra óptica e comunicação sem fio ção. O trabalho da camada de link de dados é delimitar o início e o fim dos quadros e obter de um lado para o outro com o grau de confiabilidade desejado. Um adequado o modelo deve incluir ambos como camadas separadas. O modelo TCP / IP não faz isso. Finalmente, embora os protocolos IP e TCP tenham sido cuidadosamente pensados e bem implementados, muitos dos outros protocolos eram ad hoc, geralmente produzidos por um casal de alunos de graduação trabalhando até se cansar. O protocolo as implementações foram então distribuídas gratuitamente, o que resultou em se tornarem amplamente utilizado, profundamente enraizado e, portanto, difícil de substituir. Alguns deles são um pouco

de um constrangimento agora. O protocolo de terminal virtual, TELNET, por exemplo, foi projetado para um terminal teletipo mecânico de dez caracteres por segundo. isto não sabe nada sobre interfaces gráficas de usuário e mouses. No entanto, ainda está em uso cerca de 30 anos depois.

1.5 EXEMPLO DE REDES

O assunto rede de computadores cobre muitos tipos diferentes de redes, grandes e pequenos, bem conhecidos e menos conhecidos. Eles têm objetivos diferentes, escalas e tecnologias. Nas seções a seguir, veremos alguns amplos, para se ter uma ideia da variedade que se encontra na área de redes de computadores. Começaremos com a Internet, provavelmente a rede mais conhecida, e examinaremos sua história, evolução e tecnologia. Então, vamos considerar o telefone celular rede. Tecnicamente, é bem diferente da Internet, contrastando bem com isso. Em seguida, apresentaremos o IEEE 802.11, o padrão dominante para redes sem fio LANs. Finalmente, veremos RFID e redes de sensores, tecnologias que tendem o alcance da rede para incluir o mundo físico e objetos do cotidiano.

1.5.1 A Internet

A Internet não é realmente uma rede, mas uma vasta coleção de diferentes redes que usam certos protocolos comuns e fornecem certos serviços comuns víncios. É um sistema incomum, pois não foi planejado por ninguém e não é controlado por qualquer um. Para entender melhor, vamos começar do início e ver como se desenvolveu e por quê. Para uma história maravilhosa da Internet, John O livro de Naughton (2000) é altamente recomendado. É um daqueles livros raros que não é apenas divertido de ler, mas também tem 20 páginas de *ibid.* de e *op. cit.* é para o sério historiador. Parte do material desta seção é baseado neste livro.

SEC. 1,5
EXEMPLO DE REDES

55

Claro, inúmeros livros técnicos foram escritos sobre a Internet e seus protocolos também. Para obter mais informações, consulte, por exemplo, Maufer (1999).

A ARPANET

A história começa no final dos anos 1950. No auge da Guerra Fria, os EUA DoD queria uma rede de comando e controle que pudesse sobreviver a uma guerra nuclear. Naquela época, todas as comunicações militares usavam a rede telefônica pública, que foi considerado vulnerável. A razão para essa crença pode ser deduzida Figura 1-25 (a). Aqui, os pontos pretos representam centrais de comutação telefônica, cada uma que estava conectado a milhares de telefones. Essas centrais de comutação eram, por sua vez, conectado a centrais de comutação de nível superior (centrais de pedágio), para formar um

hierarquia nacional com apenas uma pequena quantidade de redundância. A vulnerabilidade de o sistema era que a destruição de alguns postos de pedágio importantes poderia fragmentá-lo em muitas ilhas isoladas.

(uma)
Pedágio
escritório
Trocá
escritório
(b)

Figura 1-25. (a) Estrutura do sistema telefônico. (b) a proposta de Baran dis-sistema de comutação tributado.

Por volta de 1960, o DoD assinou um contrato com a RAND Corporation para encontrar um solução. Um de seus funcionários, Paul Baran, propôs o altamente distribuído e o projeto tolerante a falhas da Figura 1.25 (b). Uma vez que os caminhos entre quaisquer dois switches

escritórios eram agora muito mais longos do que os sinais analógicos podiam viajar sem distorções ção, Baran propôs o uso de tecnologia de comutação digital de pacotes. Baran escreveu sete Vários relatórios para o DoD descrevendo suas idéias em detalhes (Baran, 1964). Funcionários em o Pentágono gostou do conceito e pediu à AT&T, então a emissora nacional dos EUA monopólio do telefone, para construir um protótipo. AT&T descartou as idéias de Baran de mão. A maior e mais rica corporação do mundo não estava prestes a permitir

Página 80

56

INTRODUÇÃO INDIVÍDUO. 1

algum jovem whippersnapper lhe diz como construir um sistema telefônico. Eles disseram A rede de Baran não pôde ser construída e a ideia foi morta.

Vários anos se passaram e ainda assim o DoD não tinha um comando-e- melhor Sistema de controle. Para entender o que aconteceu a seguir, temos que voltar todos os até outubro de 1957, quando a União Soviética derrotou os EUA no espaço com o lançamento do primeiro satélite artificial, o Sputnik. Quando o presidente Eisenhower tentou descobrir quem estava dormindo no interruptor, ele ficou chocado ao encontrar o Exército, a Marinha,

e a Força Aérea disputando o orçamento de pesquisa do Pentágono. Seu imediato resposta foi criar uma única organização de pesquisa de defesa, **ARPA** , a **Agência de Projetos de Pesquisa Avançada** . O ARPA não tinha cientistas ou laboratórios; na verdade, não tinha nada mais do que um escritório e uma pequena (para os padrões do Pentágono) despesas. Ele fez seu trabalho emitindo bolsas e contratos para universidades e empresas cujas idéias pareciam promissoras.

Nos primeiros anos, a ARPA tentou descobrir qual deveria ser sua missão.

Em 1967, a atenção de Larry Roberts, um gerente de programa da ARPA que era tentando descobrir como fornecer acesso remoto a computadores, ligado à rede trabalhando. Ele contatou vários especialistas para decidir o que fazer. Um deles, Wes-Ly Clark sugeriu construir uma sub-rede comutada por pacotes, conectando cada host a seu próprio roteador.

Depois de algum ceticismo inicial, Roberts comprou a ideia e apresentou um que artigo vago sobre isso no Simpósio ACM SIGOPS sobre Sistema Operacional Princípios realizados em Gatlinburg, Tennessee, no final de 1967 (Roberts, 1967). Muito para A surpresa de Roberts, outro artigo na conferência descreveu um sistema semelhante que não foi apenas projetado, mas totalmente implementado sob a direção de Donald Davies no Laboratório Nacional de Física da Inglaterra. O sistema NPL não era um sistema nacional (apenas conectou vários computadores no NPL campus), mas demonstrou que a comutação de pacotes pode funcionar. Pele- além disso, citava o trabalho anterior agora descartado de Baran. Roberts saiu de Gatlinburg decidiu construir o que mais tarde ficou conhecido como **ARPANET** .

A sub-rede consistiria em minicomputadores chamados **IMPs (Interface Message Processadores)** conectados por linhas de transmissão de 56 kbps. Para alta confiabilidade, cada O IMP seria conectado a pelo menos dois outros IMPs. A sub-rede deveria ser um sub-rede de datagramas, então, se algumas linhas e IMPs forem destruídos, as mensagens podem ser redirecionado automaticamente ao longo de caminhos alternativos. Cada nó da rede deveria consistir em um IMP e um host, no mesmo sala, conectada por um fio curto. Um host pode enviar mensagens de até 8063 bits para seu IMP, que iria então dividi-los em pacotes de no máximo 1008 bits e encaminhe-os independentemente para o destino. Cada pacote foi recebido em sua totalidade antes de ser encaminhada, de modo que a sub-rede foi a primeira loja eletrônica

e-forward packet-switching network.

A ARPA então lançou uma licitação para construir a sub-rede. Doze empresas concorrem para isso. Depois de avaliar todas as propostas, o ARPA escolheu a BBN, consultoria com sede em Cambridge, Massachusetts, e em dezembro de 1968 concedeu-lhe um contrato

Página 81

SEC. 1,5
EXEMPLO DE REDES

57

para construir a sub-rede e escrever o software da sub-rede. BBN escolheu usar especialmente Minicomputadores Honeywell DDP-316 modificados com 12K palavras de 16 bits de núcleo memória como os IMPs. Os IMPs não tinham discos, pois as partes móveis eram considerado não confiável. Os IMPs foram interligados por linhas de 56 kbps alugadas de companhias telefônicas. Embora 56 kbps seja agora a escolha dos adolescentes que podem não pagar DSL ou cabo, era então o melhor que o dinheiro poderia comprar. O software foi dividido em duas partes: sub-rede e host. O software de sub-rede consistia na extremidade IMP da conexão host-IMP, o protocolo IMP-IMP e um protocolo IMP de origem para o IMP de destino projetado para melhorar a confiabilidade. o projeto original da ARPANET é mostrado na Figura 1-26.

Host-IMP
protocolo
Protocolo host-host
Fonte IMP para destino IMPprotocol
Protocolo IMP-IMP
IMP-IMP
protocolo
Hospedeiro
criança levada
Sub-rede

Figura 1-26. O design original da ARPANET.

Fora da sub-rede, o software também era necessário, ou seja, a extremidade do host do conexão host-IMP, o protocolo host-host e o software de aplicativo. Logo ficou claro que a BBN era de opinião que, quando aceitou uma mensagem em um fio host-IMP e colocado no fio host-IMP no destino, seu trabalho era feito.

Roberts tinha um problema: os hosts também precisavam de software. Lidar com ele, ele convocou uma reunião de pesquisadores da rede, a maioria alunos de pós-graduação, Snowbird, Utah, no verão de 1969. Os alunos de pós-graduação esperavam algum especialista em rede para explicar o grande projeto da rede e seu software para eles e então atribuir a cada um deles a tarefa de escrever parte dele. Eles ficaram surpresos quando não havia especialista em rede e nenhum grande projeto. Eles tiveram que descobrir o que fazer por conta própria.

No entanto, de alguma forma, uma rede experimental ficou online em dezembro 1969 com quatro nós: na UCLA, UCSB, SRI e na Universidade de Utah. Estes quatro foram escolhidos porque todos tinham um grande número de contratos ARPA, e todos tinham computadores host diferentes e completamente incompatíveis (apenas para torná-lo mais divertido). A primeira mensagem host a host havia sido enviada dois meses antes da UCLA

Página 82

58
INTRODUÇÃO
INDIVÍDUO. 1

nó por uma equipe liderada por Len Kleinrock (um pioneiro da teoria de comutação de pacotes) para o nó SRI. A rede cresceu rapidamente à medida que mais IMPs eram entregues e instalado; logo se espalhou pelos Estados Unidos. A Figura 1-27 mostra quão rapidamente o A ARPANET cresceu nos primeiros 3 anos.

MIT
BBN
RAND
UCLA
UCLA
SRI
UTAH
ILLINOIS MIT
LINCOLN CASE
CARN

```

HARVARD BURROUGHS
BBN
RAND
SDC
STAN
UCLA
SRI
UTAH
UCSB
SDC
UCSB
SRI
UTAH
UCSB
NCAR
GWC
LINCOLN CASE
MITRA
ETAC
HARVARD
NBS
BBN
FUNILEIRO
RAND
SDC
USC
AMES
STAN
UCLA
CARN
SRI
UTAH
MCCELLAN
UCSB
ILLINOIS
LINC
RADC
MIT
ILLINOIS
MIT
LINC
RADC
UTAH
FUNILEIRO
RAND
MCCELLAN
LBL
SKI
AMES TIP
AMES IMP
X-PARC
FWWG
FCSB
UCSD
STANFORD
CCA
BBN
HARVARD
ABERDEEN
NBS
ETAC
ARPA
MITRA
SAAC
BELVOIR
CMU
GWC
CASO
NOAA
USC
SDC
UCLA
(uma)
(d)
(b)
(c)
(e)

```

Figura 1-27. Crescimento da ARPANET. (a) Dezembro de 1969. (b) Julho de 1970.

(c) março de 1971. (d) abril de 1972. (e) setembro de 1972.

Além de ajudar a incipiente ARPANET a crescer, a ARPA também financiou pesquisa sobre a utilização de redes de satélite e redes de rádio por pacotes móveis. Em um agora famosa demonstração, um caminhão circulando na Califórnia usou o pacote rede de rádio para enviar mensagens ao SRI, que então eram encaminhadas pelo ARPANET para a Costa Leste, de onde foram enviados para o University College em Londres pela rede de satélites. Isso permitiu que um pesquisador no caminhão usasse um computador em Londres enquanto dirige pela Califórnia. Este experimento também demonstrou que os protocolos ARPANET existentes não eram adequados para funcionar em redes diferentes. Esta observação levou a mais pesquisas sobre protocolos, culminando com a invenção do modelo TCP / IP e protocolos (Cerf e Kahn, 1974). TCP / IP foi projetado especificamente para lidar com comunicação através de redes, algo se tornando cada vez mais importante à medida que mais e mais redes foram conectadas à ARPANET.

cidade da Califórnia em Berkeley reescreveu o TCP / IP com uma nova interface de programação chamados de **sockets** para o próximo lançamento 4.2BSD do Berkeley UNIX . Eles também escreveu muitos aplicativos, utilitários e programas de gerenciamento para mostrar como conveniente era usar a rede com tomadas.

O momento foi perfeito. Muitas universidades tinham acabado de adquirir um segundo ou terceiro Computador VAX e uma LAN para conectá-los, mas eles não tinham software de rede.

Quando o 4.2BSD apareceu, com TCP / IP, sockets e muitos utilitários de rede, o pacote completo foi adotado imediatamente. Além disso, com TCP / IP, foi fácil para as LANs se conectar à ARPANET, e muitos o fizeram.

Durante a década de 1980, redes adicionais, especialmente LANs, foram conectadas à ARPANET. Conforme a escala aumentou, encontrar hosts tornou-se cada vez mais caro e difícil, então **DNS** (**Sistema de Nome de Domínio**) foi criado para organizar as máquinas em domínios e mapear nomes de host em endereços IP. Desde então, o DNS se tornou um sistema de banco de dados distribuído e generalizado para armazenar uma variedade de informações relacionadas

ed para nomear. Vamos estudá-lo em detalhes no cap. 7

NSFNET

No final da década de 1970, a NSF (US National Science Foundation) viu o enorme impacto que a ARPANET estava tendo na pesquisa universitária, permitindo pesquisadores de todo o país para compartilhar dados e colaborar em projetos de pesquisa. Como nunca, para entrar na ARPANET uma universidade tinha que ter um contrato de pesquisa com o DoD. Muitos não tinham contrato. A resposta inicial da NSF foi financiar o Computer Science Network (**CSNET**) em 1981. Conectou a ciência da computação de departamentos e laboratórios de pesquisa industrial para a ARPANET via dial-up e alugado linhas. No final da década de 1980, a NSF foi mais longe e decidiu projetar um sucessor para a ARPANET que estaria aberta a todos os grupos de pesquisa universitários.

Para ter algo concreto para começar, a NSF decidiu construir um backbone rede para conectar seus seis centros de supercomputadores, em San Diego, Boulder, Cham-Paign, Pittsburgh, Ithaca e Princeton. Cada supercomputador recebeu um pouco irmão, consistindo de um microcomputador LSI-11 chamado **fuzzball** . As bolas de pelo foram conectados com linhas alugadas de 56 kbps e formaram a sub-rede, o mesmo hardware tecnologia usada pela ARPANET. A tecnologia de software era diferente no entanto: os fuzzballs falavam TCP / IP desde o início, tornando-o o primeiro WAN TCP / IP.

A NSF também financiou algumas (eventualmente cerca de 20) redes regionais que conectaram ao backbone para permitir que usuários em milhares de universidades, laboratórios de pesquisa, bibliotecas, e museus para acessar qualquer um dos supercomputadores e se comunicar com um outro. A rede completa, incluindo backbone e as redes regionais, foi chamado de **NSFNET** . Ele se conectou à ARPANET por meio de um link entre um

60

INTRODUÇÃO

INDIVÍDUO. 1

IMP e um fuzzball na sala de máquinas Carnegie-Mellon. A primeira NSFNET backbone é ilustrado na Fig. 1-28 sobreposto em um mapa dos EUA

NSF Supercomputer center

Rede NSF de nível médio

Ambos

Figura 1-28. O backbone da NSFNET em 1988.

A NSFNET foi um sucesso instantâneo e ficou sobrecarregada desde o início.

A NSF imediatamente começou a planejar seu sucessor e fechou um contrato com a Consórcio MERIT com sede em Michigan para gerenciá-lo. Canais de fibra óptica a 448 kbps foram alugados da MCI (desde a fusão com a WorldCom) para fornecer a versão 2 espinha dorsal. IBM PC-RTs foram usados como roteadores. Isso também foi logo sobrecarregado, e em 1990, o segundo backbone foi atualizado para 1,5 Mbps.

Como o crescimento continuou, a NSF percebeu que o governo não poderia continuar

financiamento de redes para sempre. Além disso, as organizações comerciais queriam ingressar, mas foram proibidos pelo contrato da NSF de usar redes pagas pela NSF. Consequentemente, a NSF incentivou MERIT, MCI e IBM a formar uma corporação sem fins lucrativos

poração, **ANS (Redes e Serviços Avançados)**, como primeiro passo ao longo do caminho para a comercialização. Em 1990, a ANS assumiu a NSFNET e atualizou o Links de 1,5 Mbps para 45 Mbps para formar **ANSNET**. Esta rede operou por 5 anos e foi então vendido para a America Online. Mas então, várias empresas foram oferecidas serviço IP comercial e estava claro que o governo agora deveria sair o negócio de rede.

Para facilitar a transição e garantir que todas as redes regionais possam comunicar cante com todas as outras redes regionais, a NSF concedeu contratos a quatro diferentes operadoras de rede para estabelecer um **NAP (Ponto de Acesso à Rede)**. Esses operadores foram PacBell (San Francisco), Ameritech (Chicago), MFS (Washington, DC), e Sprint (New York City, onde para fins de NAP, Pennsauken, New Jersey conta como cidade de Nova York). Cada operadora de rede que quisesse fornecer de volta serviço ósseo para as redes regionais NSF teve que se conectar a todos os NAPs.

Página 85

SEC. 1,5
EXEMPLO DE REDES

61

Este arranjo significava que um pacote originado em qualquer rede regional tinha uma escolha de operadoras de backbone para ir de seu NAP ao NAP de destino. Vigarista-consequêntemente, as operadoras de backbone foram forçadas a competir pela rede regional funciona o negócio com base no serviço e no preço, que era a ideia, claro.

Como resultado, o conceito de um único backbone padrão foi substituído por um comercial infraestrutura competitiva impulsionada oficialmente. Muitas pessoas gostam de criticar o Federal Governo por não ser inovador, mas na área de networking, foi o DoD e NSF que criou a infraestrutura que formou a base para a Internet e em seguida, entregou-o à indústria para operar.

Durante a década de 1990, muitos outros países e regiões também construíram redes de pesquisa, frequentemente padronizadas na ARPANET e NSFNET. Estes em incluiu EuropaNET e EBONE na Europa, que começou com linhas de 2 Mbps e depois atualizado para linhas de 34 Mbps. Eventualmente, a infraestrutura de rede em A Europa também foi entregue à indústria.

A Internet mudou muito desde aqueles primeiros dias. Explodiu em tamanho com o surgimento da World Wide Web (WWW) no início de 1990.

Dados recentes do Internet Systems Consortium colocam o número de In-hosts ternet em mais de 600 milhões. Este palpite é apenas uma estimativa baixa, mas muito excede os poucos milhões de hosts que existiam quando a primeira conferência no WWW foi realizada no CERN em 1994.

A maneira como usamos a Internet também mudou radicalmente. Inicialmente, os aplicativos como e-mail para acadêmicos, grupos de notícias, login remoto e domínio de transferência de arquivos inated. Mais tarde, mudou para o e-mail para todos, depois a Web e ponto a ponto distribuição de conteúdo, como o agora fechado Napster. Agora a mídia em tempo real distribuição, redes sociais (por exemplo, Facebook) e microblogging (por exemplo, Twitter) são decolando. Essas mudanças trouxeram tipos de mídia mais ricos para a Internet e, portanto, muito mais tráfego. Na verdade, o tráfego dominante na Internet parece mudar com alguma regularidade como, por exemplo, novas e melhores formas de trabalhar com música ou os filmes podem se tornar muito populares muito rapidamente.

Arquitetura da Internet

A arquitetura da Internet também mudou muito à medida que cresceu explosivamente. Nesta seção, tentaremos dar uma breve visão geral do que parece hoje. O quadro é complicado por contínuas turbulências no empresas de empresas de telefonia (telcos), empresas de cabo e ISPs que muitas vezes

torna difícil saber quem está fazendo o quê. Um impulsionador dessas convulsões é o setor de telecomunicações

convergência de comunicações, em que uma rede é usada para usa. Por exemplo, em um "jogo triplo", uma empresa vende telefonia, TV e Serviço de Internet na mesma conexão de rede, supondo que isso poupar dinheiro. Consequentemente, a descrição dada aqui será necessariamente um pouco mais simples do que a realidade. E o que é verdade hoje pode não ser amanhã.

Página 86

62

INTRODUÇÃO INDIVÍDUO. 1

O quadro geral é mostrado na Figura 1.29. Vamos examinar esta figura por peça, começando com um computador em casa (nas bordas da figura). Para se juntar ao Internet, o computador está conectado a um **provedor de serviços de Internet**, ou simplesmente **ISP**, de quem o usuário adquire **acesso ou conectividade à Internet**. Isso permite que o computador troca pacotes com todos os outros hosts acessíveis na Internet.

O usuário pode enviar pacotes para navegar na Web ou para qualquer um dos milhares de outros usos,

não importa. Existem muitos tipos de acesso à Internet, e geralmente são distinguidos por quanta largura de banda eles fornecem e quanto custam, mas o atributo mais importante é a conectividade.

Dados
Centro
Fibra
(FTTH)
DSL
Discar
Cabo
3G móvel
telefone
Tier 1 ISP
De outros
ISPs
Encarando
no IXP
POP
Dados
caminho
Roteador
Cabo
modem
CMTS
Espinha dorsal
DSLAM
Modem DSL

Figura 1-29. Visão geral da arquitetura da Internet.

Uma maneira comum de se conectar a um ISP é usar a linha telefônica de sua casa, em caso em que sua companhia telefônica é seu ISP. **DSL**, abreviação de **Digital Subscriber Line**, reutiliza a linha telefônica que se conecta à sua casa para dados digitais transmissão. O computador está conectado a um dispositivo chamado **modem DSL** que

converte entre pacotes digitais e sinais analógicos que podem passar desimpedidos a linha telefônica. Na outra extremidade, um dispositivo chamado **DSLAM (Digital Subscriber Line Access Multiplexer)** converte entre sinais e pacotes.

Diversas outras maneiras populares de se conectar a um ISP são mostradas na Figura 1.29. DSL é uma forma de maior largura de banda de usar a linha telefônica local do que enviar bits por um chamada telefônica tradicional em vez de uma conversa de voz. Isso é chamado **dial-up** e feito com um tipo diferente de modem em ambas as extremidades. A palavra **modem** é curta para "*mo* dulator *dem* odulator" e refere-se a qualquer dispositivo que converte entre digi-bits e sinais analógicos.

Outro método é enviar sinais pelo sistema de TV a cabo. Como DSL, este é uma forma de reutilizar a infraestrutura existente, neste caso TV a cabo não utilizada

Página 87

63

canais. O dispositivo em casa é chamado de **modem a cabo** e o dispositivo em o **headend do cabo** é chamado de **CMTS** (**Cable Modem Termination System**). DSL e cabo fornecem acesso à Internet a taxas de uma pequena fração de um megabit / seg para vários megabit / seg, dependendo do sistema. Essas taxas são muito maiores do que as taxas dial-up, que são limitadas a 56 kbps por causa da largura de banda da linha usada para chamadas de voz. Acesso à Internet muito maior do que dial-up velocidades é chamado de **banda larga**. O nome se refere à largura de banda mais ampla que é usado para redes mais rápidas, em vez de qualquer velocidade específica.

Os métodos de acesso mencionados até agora são limitados pela largura de banda do "última milha" ou última etapa da transmissão. Ao levar fibra óptica para as residências, O acesso à Internet pode ser fornecido a taxas da ordem de 10 a 100 Mbps. Isto design é denominado **FTTH** (**Fiber to the Home**). Para empresas comerciais áreas, pode fazer sentido alugar uma linha de transmissão de alta velocidade dos escritórios para o ISP mais próximo. Por exemplo, na América do Norte, uma linha T3 funciona a cerca de 45 Mbps.

Sem fio é usado para acesso à Internet também. Um exemplo que exploraremos em breve é das redes de telefonia móvel 3G. Eles podem fornecer entrega de dados a taxas de 1 Mbps ou superior para telefones celulares e assinantes fixos na área de cobertura.

Agora podemos mover pacotes entre a casa e o ISP. Nós chamamos o local ção na qual os pacotes do cliente entram na rede do ISP para atender ao **POP** do ISP (**Ponto de Presença**). A seguir, explicaremos como os pacotes são movidos entre os POPs de diferentes ISPs. Deste ponto em diante, o sistema é totalmente digital e pacote mudou.

As redes de ISP podem ser regionais, nacionais ou internacionais em escopo. Nós temos já vi que sua arquitetura é composta por linhas de transmissão de longa distância que interconectam roteadores em POPs nas diferentes cidades atendidas pelos ISPs. Isto o equipamento é chamado de **backbone** do ISP. Se um pacote é destinado a um host servido diretamente pelo ISP, esse pacote é roteado pelo backbone e entregue para o hospedeiro. Caso contrário, deve ser entregue a outro ISP.

ISPs conectam suas redes para trocar tráfego em **IXPs** (**Internet eXchange Pontos**). Os ISPs conectados são considerados **pares** entre si. Há muitos IXPs em cidades ao redor do mundo. Eles são desenhados verticalmente na Fig. 1-29 porque As redes ISP se sobrepõem geograficamente. Basicamente, um IXP é uma sala cheia de roteadores, pelo menos um por ISP. Uma LAN na sala conecta todos os roteadores, para que os pacotes possam ser encaminhado de qualquer backbone de ISP para qualquer outro backbone de ISP. IXPs podem ser instalações grandes e de propriedade independente. Um dos maiores é o Amsterdam Internet Exchange, ao qual centenas de ISPs se conectam e por meio do qual troque centenas de gigabits / s de tráfego.

O peering que acontece nos IXPs depende das relações comerciais entre entre os ISPs. Existem muitos relacionamentos possíveis. Por exemplo, um pequeno ISP pode pagar um ISP maior para conectividade com a Internet para alcançar hosts distantes, assim como um o cliente adquire o serviço de um provedor de Internet. Neste caso, o pequeno ISP diz-se que paga pelo **trânsito**. Como alternativa, dois grandes ISPs podem decidir trocar

64

ing (Metz, 2001).

O caminho que um pacote percorre na Internet depende das opções de peering de os ISPs. Se o ISP que entrega um pacote faz o mesmo com o ISP de destino, ele pode entregar o pacote diretamente ao seu par. Caso contrário, ele pode rotear o pacote para o lugar mais próximo em que se conecta a um provedor de transporte público pago para que o provedor possa

entregar o pacote. Dois exemplos de caminhos entre os ISPs são desenhados na Figura 1.29. Frequentemente,

o caminho que um pacote segue não será o caminho mais curto pela Internet.

No topo da cadeia alimentar está um pequeno punhado de empresas, como AT&T e Sprint, que opera grandes redes de backbone internacionais com milhares de rotas conectados por links de fibra óptica de alta largura de banda. Esses ISPs não pagam por transito. Eles são geralmente chamados **de ISPs de nível 1** e formam a espinha dorsal da Internet, uma vez que todos os outros devem se conectar a eles para poderem alcançar o pneu Internet.

Empresas que fornecem muito conteúdo, como Google e Yahoo !, localizam seus computadores em **datacenters** bem conectados ao restante da Internet.

Esses data centers são projetados para computadores, não humanos, e podem ser preenchidos com rack sobre rack de máquinas chamadas de **server farm**. **Colocação ou hospedagem** data centers permitem que os clientes coloquem equipamentos como servidores em ISP POPs para que

curtas, conexões rápidas podem ser feitas entre os servidores e os backbones do ISP.

A indústria de hospedagem na Internet tem se tornado cada vez mais virtualizada, de modo que agora é

comum alugar uma máquina virtual que é executada em um farm de servidores em vez de instalar um computador físico. Esses data centers são tão grandes (dezenas ou centenas de milhares de máquinas) que a eletricidade é um custo importante, então os data centers são algumas vezes construído em áreas onde a eletricidade é barata.

Isso encerra nosso rápido tour pela Internet. Teremos muito a dizer sobre os componentes individuais e seu design, algoritmos e protocolos em capítulos subsequentes. Outro ponto que vale a pena mencionar aqui é que significa estar na Internet está mudando. Costumava ser que uma máquina estava no Internet se: (1) executou a pilha de protocolos TCP / IP; (2) tinha um endereço IP; e (3) poderia enviar pacotes IP para todas as outras máquinas na Internet. No entanto, ISPs costumam reutilizar endereços IP, dependendo de quais computadores estão em uso no momento, e as redes domésticas geralmente compartilham um endereço IP entre vários computadores. Isto a prática mina a segunda condição. Medidas de segurança, como firewalls também pode bloquear parcialmente os computadores de receber pacotes, prejudicando o terceiro ponto. Apesar dessas dificuldades, faz sentido considerar essas máquinas como estar na Internet enquanto estão conectados aos seus ISPs.

Também vale a pena mencionar de passagem é que algumas empresas se interconectaram todas as suas redes internas existentes, muitas vezes usando a mesma tecnologia que o Internet. Essas **intranets** são normalmente acessíveis apenas nas instalações da empresa ou de notebooks da empresa, mas funcionam da mesma maneira que a Internet.

Página 89

SEC. 1,5
EXEMPLO DE REDES

65

1.5.2 Redes de telefonia móvel de terceira geração

As pessoas gostam mais de falar ao telefone do que de navegar na Internet, e isso tornou a rede de telefonia móvel a rede de maior sucesso no mundo. Possui mais de quatro bilhões de assinantes em todo o mundo. Para colocar este número em perspectiva, é cerca de 60% da população mundial e mais do que o número de hosts da Internet e linhas de telefone fixo combinados (ITU, 2009).

A arquitetura da rede de telefonia móvel mudou muito ao longo dos últimos 40 anos junto com seu tremendo crescimento. Celular de primeira geração

sistemas transmitiam chamadas de voz como sinais de variação contínua (análogicos), em vez do que sequências de bits (digitais). **AMPS** (**Sistema Avançado de Telefonia Móvel**), que foi implantado nos Estados Unidos em 1982, foi amplamente utilizado pela primeira vez sistema de geração. Sistemas de telefonia móvel de segunda geração mudados para transmitting chamadas de voz em formato digital para aumentar a capacidade, melhorar a segurança e oferecer

mensagem de texto. **GSM** (**Sistema Global para Comunicações Móveis**), que foi implantado a partir de 1991 e se tornou o telefone móvel mais usado sistema do mundo, é um sistema 2G.

Os sistemas de terceira geração, ou 3G, foram inicialmente implantados em 2001 e oferecem serviços de voz digital e de dados digitais em banda larga. Eles também vêm com muitos de jargão e muitos padrões diferentes para escolher. 3G é vagamente definido por a ITU (um organismo de padrões internacionais que discutiremos na próxima seção) como fornecendo taxas de pelo menos 2 Mbps para usuários estacionários ou caminhando e 384 kbps em um veículo em movimento. **UMTS** (**Sistema Universal de Telecomunicações Móveis**), também chamado de **WCDMA** (**Wideband Code Division Multiple Access**), é o principal Sistema 3G que está sendo rapidamente implantado em todo o mundo. Pode fornecer até 14 Mbps no downlink e quase 6 Mbps no uplink. Versões futuras usarão várias antenas e rádios para fornecer velocidades ainda maiores para os usuários.

O recurso escasso em sistemas 3G, como nos sistemas 2G e 1G antes deles, é espectro de rádio. Os governos licenciam o direito de usar partes do espectro para o operadoras de rede de telefonia móvel, muitas vezes usando um leilão de espectro em que rede os operadores apresentam licitações. Ter um pedaço de espectro licenciado torna mais fácil decifrar assinar e operar sistemas, uma vez que ninguém mais pode transmitir nesse espectro, mas geralmente custa muito dinheiro. No Reino Unido em 2000, por exemplo, cinco Licenças 3G foram leiloadas por um total de cerca de US \$ 40 bilhões.

É a escassez de espectro que levou ao design da **rede celular** mostrado em Figura 1-30, que agora é usada para redes de telefonia móvel. Para gerenciar o rádio interferência entre os usuários, a área de cobertura é dividida em células. Dentro de uma célula, os usuários são atribuídos a canais que não interferem uns com os outros e não causam muita interferência para células adjacentes. Isso permite uma boa reutilização da especificação trum, ou **reutilização de frequência**, nas células vizinhas, o que aumenta a capacidade da rede. Em sistemas 1G, que transportava cada chamada de voz em uma freqüência específica banda de frequência, as freqüências foram cuidadosamente escolhidas para que não entrassem em conflito com células vizinhas. Desta forma, uma determinada freqüência pode ser reutilizada apenas uma vez

Página 90

66

INTRODUÇÃO INDIVÍDUO. 1

em várias células. Os modernos sistemas 3G permitem que cada célula use todas as freqüências, mas em uma forma que resulta em um nível tolerável de interferência para as células vizinhas. Existem variações no design celular, incluindo o uso de direcional ou sec- antenas em torres de celular para reduzir ainda mais a interferência, mas a ideia básica é o mesmo.

Células
Estação base

Figura 1-30. Desenho celular de redes de telefonia móvel.

A arquitetura da rede de telefonia móvel é muito diferente daquela do Internet. Possui várias partes, conforme mostrado na versão simplificada do UMTS ar- arquitetura na Figura 1-31. Primeiro, existe a **interface aérea**. Este termo é um nome chique para o protocolo de comunicação de rádio que é usado pelo ar entre o celular dispositivo (por exemplo, o telefone celular) e a **estação base de celular**. Avanços no ar em superfície nas últimas décadas aumentou muito as taxas de dados sem fio. o A interface aérea UMTS é baseada em **Code Division Multiple Access** (**CDMA**), uma tecnologia que estudaremos no cap. 2

A estação de base celular junto com seu controlador forma o **acesso de rádio rede**. Esta parte é o lado sem fio da rede de telefonia móvel. O con-nó controlador ou **RNC (controlador de rede de rádio)** controla como o espectro é usava. A estação base implementa a interface aérea. É chamado de **Nó B**, um temporizador rótulo porary que pegou.

O resto da rede de telefonia móvel transporta o tráfego para o acesso de rádio rede. É chamada de **rede principal**. A rede central UMTS evoluiu de a rede central usada para o sistema GSM 2G anterior. Contudo, algo surpreendente está acontecendo na rede central UMTS.

Desde o início da rede, uma guerra está acontecendo entre as pessoas pessoas que suportam redes de pacotes (ou seja, sub-redes sem conexão) e as pessoas que redes de circuito de suporte (ou seja, sub-redes orientadas para conexão). Os principais proponentes de pacotes vêm da comunidade da Internet. Em um design sem conexão, cada o pacote é roteado independentemente de qualquer outro pacote. Como consequência, se algum roteadores caem durante uma sessão, nenhum dano será causado, desde que o sistema possa

Página 91

SEC. 1,5
EXEMPLO DE REDES

67

RNC
RNC
MSC /
MGW
GMSC
/ MGW
SGSN
GGSN
Rede de acesso de rádio
Rede central
Ar
interface
("Uu")
Nó B
PSTN
Internet
Pacotes
Circuitos
("Iu-CS")
Acesso
/ Testemunho
interface
("Iu")
Pacotes
("Iu-PS")
HSS

Figura 1-31. Arquitetura da rede de telefonia móvel UMTS 3G.

reconfigurar-se dinamicamente para que os pacotes subsequentes possam encontrar alguma rota para o destino, mesmo que seja diferente daquele usado pelos pacotes anteriores.

O campo de circuito vem do mundo das companhias telefônicas. No tele-sistema telefônico, um chamador deve discar o número da parte chamada e esperar por uma conexão antes de falar ou enviar dados. Esta configuração de conexão estabelece uma rota através do sistema telefônico que é mantido até que a chamada seja encerrada. Todos palavras ou pacotes seguem a mesma rota. Se uma linha ou interruptor no caminho cair, a chamada é abortada, tornando-a menos tolerante a falhas do que um design sem conexão.

A vantagem dos circuitos é que eles podem suportar a qualidade de serviço com mais facilidade ly. Ao configurar uma conexão com antecedência, a sub-rede pode reservar recursos como como largura de banda do link, espaço de buffer de switch e CPU. Se for feita uma tentativa de configurar

uma chamada e recursos insuficientes estão disponíveis, a chamada é rejeitada e o chamador recebe uma espécie de sinal de ocupado. Desta forma, uma vez que uma conexão foi estabelecida, o conexão obterá um bom serviço.

Com uma rede sem conexão, se muitos pacotes chegarem ao mesmo roteador ao mesmo tempo, o roteador irá engasgar e provavelmente perder os pacotes. O remetente eventualmente perceberá isso e os reenviará, mas a qualidade do serviço será instável e inadequados para áudio ou vídeo, a menos que a rede esteja levemente carregada. Desnecessário

digamos, fornecer qualidade de áudio adequada é algo que as empresas telefônicas se preocupam sobre muito, daí sua preferência por conexões.

A surpresa na Figura 1-31 é que há tanto pacotes quanto circuitos comutados equipamento na rede principal. Isso mostra a rede de telefonia móvel em transição ção, com as empresas de telefonia móvel capazes de implementar um ou às vezes ambos

Página 92

68

INTRODUÇÃO

INDIVÍDUO. 1

as alternativas. Redes de telefonia móvel mais antigas usavam um núcleo comutado por circuito no estilo da rede telefônica tradicional para realizar chamadas de voz. Este legado é visto em a rede UMTS com o **MSC** (**Mobile Switching Center**), **GMSC** (**Gate-forma Mobile Switching Center**), e elementos **MGW** (**Media Gateway**) que definem conexões através de uma rede central comutada por circuito, como a **PSTN** (**pública Rede telefônica comutada**).

Os serviços de dados se tornaram uma parte muito mais importante do telefone móvel rede do que costumavam ser, começando com mensagens de texto e primeiros pacotes de dados serviços como **GPRS** (**General Packet Radio Service**) no sistema GSM.

Esses serviços de dados mais antigos eram executados a dezenas de kbps, mas os usuários queriam mais. Novo mo-

as redes de telefonia biliar transportam dados em pacotes a taxas de vários Mbps. Para comparação, uma chamada de voz é transportada a uma taxa de 64 kbps, normalmente 3-4x menos com compressão.

Para transportar todos esses dados, os nós da rede central UMTS se conectam diretamente a um rede comutada por pacotes. O **SGSN** (**Serving GPRS Support Node**) e o **GGSN** (**Gateway GPRS Support Node**) entrega pacotes de dados de e para celulares e interface com redes de pacotes externos, como a Internet.

Esta transição deve continuar nas redes de telefonia móvel que agora estão sendo planejado e implantado. Os protocolos da Internet são usados até mesmo em celulares para configurar

conexões para chamadas de voz em uma rede de pacote de dados, na forma de voz-sobre IP. IP e pacotes são usados desde o acesso de rádio até o rede central. Claro, a maneira como as redes IP são projetadas também está mudando para apoiar uma melhor qualidade de serviço. Se não, problemas com picados o áudio e o vídeo irregular não impressionariam os clientes pagantes. Voltaremos a isso assunto no cap. 5

Outra diferença entre as redes de telefonia móvel e o tradicional Inter net é mobilidade. Quando um usuário sai do alcance de uma estação base de celular e no intervalo de outro, o fluxo de dados deve ser reencaminhado do antigo para a nova estação de base celular. Esta técnica é conhecida como **handover** ou **handoff**, e é ilustrado na Figura 1-32.

(uma)
(b)

Figura 1-32. Transferência do telefone móvel (a) antes, (b) depois.

Tanto o dispositivo móvel quanto a estação base podem solicitar uma transferência quando o qualidade do sinal cai. Em algumas redes de celular, geralmente aquelas baseadas em CDMA

Página 93

SEC. 1,5

EXEMPLO DE REDES

69

tecnologia, é possível conectar à nova estação base antes de desconectar da antiga estação base. Isso melhora a qualidade da conexão para o celular ser-porque não há interrupção no serviço; o celular está realmente conectado a duas bases estações por um curto período. Esta maneira de fazer uma transferência é chamada **de transferência suave**

para distingui-lo de um **handover rígido**, em que o celular se desconecta do antiga estação base antes de conectar-se à nova.

Um problema relacionado é como encontrar um celular em primeiro lugar quando há um próxima chamada. Cada rede de telefonia móvel possui um **HSS (Home Subscriber Server)** na rede central que conhece a localização de cada assinante, bem como de outros informações de perfil que são usadas para autenticação e autorização. Nesse caminho, cada móvel pode ser encontrado entrando em contato com o HSS.

Uma área final a ser discutida é a segurança. Historicamente, as empresas de telefonia adotaram segurança muito mais a sério do que as empresas de Internet por muito tempo por causa de a necessidade de faturar pelo serviço e evitar fraudes (de pagamento). Infelizmente isso não é falando muito. No entanto, na evolução das tecnologias 1G para 3G, empresas de telefonia móvel foram capazes de lançar alguns mecanismos básicos de segurança ismos para celulares.

Começando com o sistema 2G GSM, o celular foi dividido em um aparelho e um chip removível contendo a identidade e a conta do assinante em formação. O chip é informalmente chamado de **cartão SIM**, abreviação de **Assinante Módulo de identidade**. Os cartões SIM podem ser trocados para telefones diferentes para ativar eles, e eles fornecem uma base para a segurança. Quando os clientes GSM viajam para outro países em férias ou negócios, muitas vezes trazem seus aparelhos, mas compram um novo Cartão SIM por alguns dólares na chegada para fazer chamadas locais sem roaming cobranças.

Para reduzir a fraude, as informações dos cartões SIM também são utilizadas pelo celular rede para autenticar assinantes e verificar se eles têm permissão para usar a rede trabalhos. Com UMTS, o celular também usa as informações do cartão SIM para verifique se ele está se comunicando com uma rede legítima.

Outro aspecto da segurança é a privacidade. Os sinais sem fio são transmitidos para todos receptores próximos, para dificultar a escuta de conversas, cripto-as chaves gráficas no cartão SIM são usadas para criptografar as transmissões. Esta abordagem oferece muito melhor privacidade do que em sistemas 1G, que eram facilmente acessados, mas não é uma panacéia devido a deficiências nos esquemas de criptografia.

As redes de telefonia móvel estão destinadas a desempenhar um papel central nas redes futuras. Agora eles estão mais voltados para aplicativos de banda larga móvel do que para chamadas de voz, e isso

tem implicações importantes para as interfaces aéreas, arquitetura de rede central e seguran a ridade das futuras redes. As tecnologias 4G que são mais rápidas e melhores estão em jogo-placa de integração com o nome de **LTE (Long Term Evolution)**, mesmo como design 3G e a implantação continua. Outras tecnologias sem fio também oferecem banda larga In-acesso ternet a clientes fixos e móveis, nomeadamente redes 802.16 sob a nome do **WiMAX**. É inteiramente possível que LTE e WiMAX estejam em uma col-O curso de ligação um com o outro e é difícil prever o que vai acontecer com eles.

1.5.3 LANs sem fio: 802.11

Quase assim que os laptops apareceram, muitas pessoas sonharam entrando em um escritório e tendo seu laptop conectado a a Internet. Consequentemente, vários grupos começaram a trabalhar em maneiras de realizar este objetivo. A abordagem mais prática é equipar o escritório e o laptop computadores com transmissores e receptores de rádio de curto alcance para permitir que eles falem. O trabalho neste campo levou rapidamente a LANs sem fio sendo comercializadas por uma variedade de empresas. O problema era que nenhum deles era compatível. O proliferação de padrões significava que um computador equipado com um rádio da marca X não trabalhar em uma sala equipada com uma estação base da marca Y. Em meados de 1990, a indústria tentou decidir que um padrão de LAN sem fio pode ser uma boa ideia, então o IEEE com

o comitê que padronizou LANs com fio recebeu a tarefa de elaborar um menor padrão LAN.

A primeira decisão foi a mais fácil: como chamá-la. Todos os outros padrões de LAN tinha números como 802.1, 802.2 e 802.3, até 802.10, então o padrão da LAN sem fio dard foi apelidado de 802.11. Um nome de gíria comum para isso é **WiFi**, mas é importante padrão importante e merece respeito, por isso vamos chamá-lo pelo seu nome próprio, 802.11. O resto foi mais difícil. O primeiro problema foi encontrar uma banda de frequência adequada que estava disponível, de preferência em todo o mundo. A abordagem adotada foi o oposto de que é usado em redes de telefonia móvel. Em vez de espectro licenciado caro, Os sistemas 802.11 operam em bandas não licenciadas, como o **ISM (Industrial, Scientificas e médicas)** definidas por ITU-R (por exemplo, 902-928 MHz, 2,4-2,5 GHz, 5,725-5,825 GHz). Todos os dispositivos estão autorizados a usar este espectro, desde que eles limitam sua potência de transmissão para permitir a coexistência de diferentes dispositivos. Claro, isso

significa que rádios 802.11 podem competir com telefones sem fio, abridores de portas de garagem e fornos de microondas.

As redes 802.11 são compostas de clientes, como laptops e telefones celulares, e infraestrutura denominada **APs (pontos de acesso)** que é instalada em edifícios. Acesso os pontos às vezes são chamados de **estações base**. Os pontos de acesso se conectam à rede com fio rede e toda a comunicação entre clientes passa por um ponto de acesso. isto também é possível para clientes que estão ao alcance do rádio falarem diretamente, como dois computadores em um escritório sem um ponto de acesso. Este arranjo é chamado de **ad hoc rede**. É usado com muito menos frequência do que o modo de ponto de acesso. Ambos os modos são

mostrado na Figura 1-33.

A transmissão 802.11 é complicada por condições sem fio que variam conforme pequenas mudanças no ambiente. Nas frequências usadas para 802.11, sinal de rádio nais podem ser refletidos em objetos sólidos para que múltiplos ecos de uma transmissão pode alcançar um receptor por caminhos diferentes. Os ecos podem cancelar ou reforçar um ao outro, fazendo com que o sinal recebido flutue muito. Este fenômeno é chamado de **desvanecimento multipercorso**, e é mostrado na Figura 1.34.

A ideia chave para superar as condições sem fio variáveis é a **diversidade de caminhos**, ou o envio de informações por caminhos múltiplos e independentes. Desta forma, o

Página 95

SEC. 1,5 EXEMPLO DE REDES

71

(uma)

(b)

Para rede com fio

Acesso

ponto

Figura 1-33. (a) Rede sem fio com um ponto de acesso. (b) Rede ad hoc.

é provável que as informações sejam recebidas mesmo que um dos caminhos seja ruim devido a um fade. Esses caminhos independentes são normalmente integrados aos módulos digitais esquema de instalação na camada física. As opções incluem o uso de diferentes frequências a-cruzar a banda permitida, seguindo diferentes caminhos espaciais entre diferentes pares de antenas, ou bits repetidos em diferentes períodos de tempo.

Sinal desbotado

Refletor

Sem fio

transmissor

Sinal não atenuado

Caminhos múltiplos

Sem fio

receptor

Figura 1-34. Desvanecimento de multipercorso.

Diferentes versões de 802.11 usaram todas essas técnicas. A inicial (1997) padrão definiu uma LAN sem fio que funcionava a 1 Mbps ou 2 Mbps por pular entre as frequências ou espalhar o sinal pelo espectro permitido.

Quase imediatamente, as pessoas reclamaram que era muito lento, então o trabalho começou

padrões mais rápidos. O projeto de espalhamento de espectro foi estendido e se tornou o (1999) padrão 802.11b rodando a taxas de até 11 Mbps. O 802.11a (1999) e Os padrões 802.11g (2003) mudaram para um esquema de modulação diferente chamado **OFDM (Multiplexação por Divisão de Freqüência Ortogonal)**. Ele divide uma ampla banda do espectro em muitas fatias estreitas sobre as quais bits diferentes são enviados em paralelo. Este esquema melhorado, que estudaremos no Cap. 2, aumentou o bit 802.11a / g

Página 96

72

INTRODUÇÃO

INDIVÍDUO. 1

taxas de até 54 Mbps. É um aumento significativo, mas as pessoas ainda queriam mais rendimento para suportar usos mais exigentes. A versão mais recente é 802.11n (2009). Ele usa bandas de frequência mais amplas e até quatro antenas por computador para alcançar taxas de até 450 Mbps.

Como o wireless é inherentemente um meio de transmissão, os rádios 802.11 também precisam lidar com o problema de que várias transmissões são enviadas ao mesmo tempo colidirão, o que pode interferir na recepção. Para lidar com este problema, 802.11 usa um esquema **CSMA (Carrier Sense Multiple Access)** que se baseia em ideias de Ethernet com fio clássica, que, ironicamente, foi extraída de uma rede sem fio antiga desenvolvida no Havaí e denominado **ALOHA**. Computadores esperam por um curto aleatório intervalo antes de transmitir, e adiar suas transmissões se ouvirem que algum outro já está transmitindo. Este esquema torna menos provável que dois computadores enviarão ao mesmo tempo. Não funciona tão bem quanto no caso de conexão com fio

redes, no entanto. Para ver por quê, examine a Figura 1-35. Suponha que o computador *A* seja transmitindo para o computador *B*, mas o alcance do rádio de *um* transmissor ‘s é muito curta para atingir computador *C*. Se *C* quiser transmitir para *B*, ele pode ouvir antes de começar, mas o fato de não ouvir nada não significa que sua transmissão será ter sucesso. A incapacidade de *C* de ouvir *A* antes de começar faz com que algumas colisões ocorram cur. Após qualquer colisão, o remetente espera outro, mais longo, atraso aleatório e retransmite o pacote. Apesar deste e de alguns outros problemas, o esquema funciona bem o suficiente na prática.

UMA
C
B
Alcance
como de
rádio
Alcance
de C's
rádio

Figura 1-35. O alcance de um único rádio pode não cobrir todo o sistema.

Outro problema é o da mobilidade. Se um cliente móvel for movido de o ponto de acesso que está usando e ao alcance de um ponto de acesso diferente, de alguma forma de entregá-lo é necessário. A solução é que uma rede 802.11 pode consistir em várias células, cada uma com seu próprio ponto de acesso e um sistema de distribuição que conecta as células. O sistema de distribuição é frequentemente comutado Ethernet, mas pode usar qualquer tecnologia. Conforme os clientes se movem, eles podem encontrar outro ponto de acesso com um sinal melhor do que o que eles estão usando atualmente e alterar sua associação. Visto de fora, todo o sistema parece uma única LAN com fio.

Página 97

SEC. 1,5

EXEMPLO DE REDES

73

Dito isso, a mobilidade em 802.11 tem sido de valor limitado até agora em comparação com mobilidade na rede móvel. Normalmente, 802.11 é usado por clientes que vão de um local fixo para outro, em vez de serem usados em trânsito.

A mobilidade não é realmente necessária para o uso nômade. Mesmo quando a mobilidade 802.11 é usado, ele se estende por uma única rede 802.11, que pode cobrir no máximo uma grande construção. Esquemas futuros precisarão fornecer mobilidade em redes diferentes e em diferentes tecnologias (por exemplo, 802.21).

Finalmente, existe o problema da segurança. Uma vez que as transmissões sem fio são transmissão, é fácil para os computadores próximos receberem pacotes de informações que não foram destinados a eles. Para evitar isso, o padrão 802.11 incluiu um en- esquema de criptografia conhecido como **WEP** (**Wired Equivalent Privacy**). A ideia era tornar a segurança sem fio como a segurança com fio. É uma boa ideia, mas não- felizmente, o esquema falhou e logo foi quebrado (Borisov et al., 2001). Tem desde então, foi substituído por esquemas mais novos que têm diferentes detalhes criptográficos no padrão 802.11i, também denominado **WiFi Protected Access**, inicialmente denominado **WPA** mas agora substituído por **WPA2**.

O 802.11 causou uma revolução nas redes sem fio que deve continuar.

Além de edifícios, está começando a ser instalado em trens, aviões, barcos e automóveis. biles para que as pessoas possam navegar na Internet onde quer que estejam. Telefones celulares e tudo

forma de eletrônicos de consumo, de consoles de jogos a câmeras digitais, podem comunicar com ele. Voltaremos a ele em detalhes no Cap. 4 -

1.5.4 RFID e redes de sensores

As redes que estudamos até agora são compostas de dispositivos de computação que são fáceis de reconhecer, de computadores a telefones celulares. Com **radiofrequência IDentification** (**RFID**), objetos do cotidiano também pode ser parte de uma rede de computadores.

Uma etiqueta RFID parece um adesivo do tamanho de um selo postal que pode ser afixado (ou embutido em) um objeto para que possa ser rastreado. O objeto pode ser uma vaca, um passaporte, livro ou palete de transporte. A etiqueta consiste em um pequeno microchip com um identificador único e uma antena que recebe as transmissões de rádio. Leitores RFID instalado em pontos de rastreamento, encontre etiquetas quando elas entrarem no alcance e interroguem

para suas informações, conforme mostrado na Figura 1-36. Os aplicativos incluem verificação identidades, gerenciando a cadeia de suprimentos, cronometrando corridas e substituindo códigos de barras.

Existem muitos tipos de RFID, cada um com propriedades diferentes, mas talvez o aspecto mais fascinante da tecnologia RFID é que a maioria das etiquetas RFID não tem um plugue elétrico nem uma bateria. Em vez disso, toda a energia necessária para operá-los é fornecido na forma de ondas de rádio por leitores RFID. Esta tecnologia é chamada **RFID passivo** para distingui-lo do (menos comum) **RFID ativo** em que há uma fonte de alimentação na tag.

Uma forma comum de RFID é **UHF RFID** (**Ultra-High Frequency RFID**). É usado em paletes de transporte e algumas carteiras de motorista. Leitores enviam sinais em

Figura 1-36. RFID usado para conectar objetos do dia a dia.

a banda 902-928 MHz nos Estados Unidos. As tags se comunicam a distâncias de vários medidores, alterando a maneira como refletem os sinais do leitor; o leitor é capaz de captar esses reflexos. Essa forma de operação é chamada de **retroespalhamento**.

Outro tipo popular de RFID é **HF RFID** (**High Frequency RFID**). isto opera a 13,56 MHz e é provável que esteja em seu passaporte, cartões de crédito, livros, e sistemas de pagamento sem contato. HF RFID tem um alcance curto, normalmente um medidor ou menos, porque o mecanismo físico é baseado na indução ao invés de espalhar. Existem também outras formas de RFID usando outras frequências, como **LF**

RFID (**RFID de baixa frequência**), que foi desenvolvido antes do HF RFID e usado para rastreamento de animais. É o tipo de RFID que provavelmente existe em seu gato. Os leitores RFID devem de alguma forma resolver o problema de lidar com várias etiquetas dentro do alcance de leitura. Isso significa que uma tag não pode simplesmente responder quando ouve um leitor ou os sinais de várias tags podem colidir. A solução é semelhante a a abordagem adotada em 802.11: as tags aguardam um curto intervalo aleatório antes de voltar a respondendo com sua identificação, o que permite ao leitor restringir tags duplas e interrogá-los ainda mais. A segurança é outro problema. A capacidade dos leitores RFID de rastrear facilmente um objeto jeto e, portanto, a pessoa que o usa, pode ser uma invasão de privacidade. Infelizmente, é difícil proteger as etiquetas RFID porque lhes falta o cálculo e poder de comunicação para executar algoritmos criptográficos robustos. Em vez disso, fraco medidas como senhas (que podem ser facilmente quebradas) são usadas. Se uma identidade cartão pode ser lido remotamente por um oficial de fronteira, o que é impedir o mesmo cartão de ser rastreado por outras pessoas sem o seu conhecimento? Não muito. As etiquetas RFID começaram como chips de identificação, mas estão rapidamente se transformando computadores desenvolvidos. Por exemplo, muitos tags têm memória que pode ser atualizada e posteriormente consultado, de modo que as informações sobre o que aconteceu com o objeto marcado pode ser armazenado com ele. Rieback et al. (2006) demonstrou que isso significa que todos dos problemas usuais de malware de computador se aplicam, apenas agora seu gato ou seu passaporte pode ser usado para espalhar um vírus RFID. Um avanço na capacidade do RFID é a **rede de sensores**. Redes de sensores são implantados para monitorar aspectos do mundo físico. Até agora, eles têm principalmente sido usado para experimentação científica, como monitoramento de habitats de pássaros, volatilidade canadense e migração zebra, mas aplicativos de negócios, incluindo saúde,

Página 99

SEC. 1,5
EXEMPLO DE REDES

75

equipamentos de monitoramento de vibração e rastreamento de congelados, refrigerados ou outros bens perecíveis sábios não podem estar muito atrás. Os nós sensores são pequenos computadores, muitas vezes do tamanho de um chaveiro, que têm temperatura, vibração e outros sensores. Muitos nós são colocados no ambiente que deve ser monitorado. Normalmente, eles têm baterias, embora possam limpar energia de vibrações ou do sol. Tal como acontece com RFID, ter energia suficiente é uma chave desafio, e os nós devem se comunicar com cuidado para serem capazes de entregar seus informações do sensor para um ponto de coleta externo. Uma estratégia comum é para os nós se auto-organizam para retransmitir mensagens uns para os outros, como mostra a Figura 1.37.

Este projeto é chamado de **rede multihop**.

Dados
coleção
ponto
Sensor
nó
Sem fio
pulo

Figura 1-37. Topologia multihop de uma rede de sensores.

RFID e redes de sensores são susceptíveis de se tornarem muito mais capazes e pervasivo no futuro. Os pesquisadores já combinaram o melhor de ambas as tecnologias gies por prototipagem de tags RFID programáveis com luz, movimento e outros sensores (Sample et al., 2008).

1.6 PADRONIZAÇÃO DA REDE

Existem muitos fornecedores e vendedores de rede, cada um com suas próprias ideias de como coisas devem ser feitas. Sem coordenação, haveria um caos completo e

os usuários não fariam nada. A única saída é concordar com alguma rede padrões. Os bons padrões não apenas permitem que diferentes computadores se comuniquem, mas também aumentam o mercado de produtos que aderem aos padrões. Um maior mercado leva à produção em massa, economias de escala na manufatura, melhor im-complementações e outros benefícios que diminuem o preço e aumentam ainda mais o aceitação.

Nesta seção, daremos uma olhada rápida no importante, mas pouco conhecido, mundo da padronização internacional. Mas vamos primeiro discutir o que pertence a um

Página 100

76

INTRODUÇÃO

INDIVÍDUO. 1

padrão. Uma pessoa razoável pode presumir que um padrão diz a você como um profissional o tocol deve funcionar para que você possa fazer um bom trabalho de implementação. Aquela pessoa estaria errado.

Os padrões definem o que é necessário para a interoperabilidade: nem mais, nem menos. que permite que o mercado mais amplo surja e também permite que as empresas competam com base em quão bons são seus produtos. Por exemplo, o padrão 802.11 define muitos taxas de transmissão, mas não diz quando um remetente deve usar qual taxa, que é um fator chave para um bom desempenho. Isso depende de quem faz o produto.

Freqüentemente, chegar à interoperabilidade dessa maneira é difícil, uma vez que existem muitos escolhas e padrões de mentação geralmente definem muitas opções. Para 802.11, há eram tantos problemas que, em uma estratégia que se tornou prática comum, um grupo comercial chamado **WiFi Alliance** começou a trabalhar na interoperabilidade com no padrão 802.11.

Da mesma forma, um padrão de protocolo define o protocolo durante a transmissão, mas não o interface de serviço dentro da caixa, exceto para ajudar a explicar o protocolo. Serviço real interfaces geralmente são proprietárias. Por exemplo, a maneira como o TCP faz a interface com o IP dentro

um computador não importa para falar com um host remoto. Só importa que o re-O host mote fala TCP / IP. Na verdade, TCP e IP são comumente implementados juntos ela sem qualquer interface distinta. Dito isso, boas interfaces de serviço, como boas APIs são valiosas para fazer com que os protocolos sejam usados e os melhores (como Berkeley soquetes) podem se tornar muito populares.

Os padrões se enquadram em duas categorias: de fato e de jure. **De facto** (latim para " do fato ") padrões são aqueles que acabaram de acontecer, sem qualquer formal plano. HTTP, o protocolo no qual a Web é executado, começou a vida como um padrão de fato dard. Fazia parte dos primeiros navegadores WWW desenvolvidos por Tim Berners-Lee em CERN, e seu uso decolou com o crescimento da web. Bluetooth é outro ex-amplo. Foi originalmente desenvolvido pela Ericsson, mas agora todos estão usando.

Padrões de **jure** (latim para "por lei"), em contraste, são adotados por meio do regras de algum organismo formal de normalização. Autorização de padronização internacional laços são geralmente divididos em duas classes: aqueles estabelecidos por tratado entre governos nacionais e aqueles que compreendem organizações não-conveniadas.

Na área de padrões de redes de computadores, existem várias organizações de cada tipo, notavelmente ITU, ISO, IETF e IEEE, todos os quais discutiremos a seguir.

Na prática, as relações entre padrões, empresas e padrões os corpos de dardização são complicados. Padrões de fato muitas vezes evoluem para de jure padrões, especialmente se forem bem-sucedidos. Isso aconteceu no caso do HTTP, que foi rapidamente escolhido pela IETF. Organismos de padrões freqüentemente ratificam uns aos outros ' padrões, no que parece ser dar tapinhas nas costas uns dos outros, para aumentar o mercado para uma tecnologia. Hoje em dia, muitas alianças de negócios ad hoc que são formado em torno de tecnologias específicas também desempenham um papel significativo no desenvolvimento e refinar os padrões de rede. Por exemplo, **3GPP (Terceira Geração**

Projeto de Parceria) é uma colaboração entre associações de telecomunicações que impulsiona os padrões de telefonia móvel UMTS 3G.

Página 101

SEC. 1,6
PADRONIZAÇÃO DA REDE

77

1.6.1 Quem é quem no mundo das telecomunicações

O status legal das companhias telefônicas mundiais varia consideravelmente de país para país. Em um extremo estão os Estados Unidos, que tem mais de 2.000 sete arate, companhias telefônicas privadas (principalmente muito pequenas). Um pouco mais foram adicionados com a dissolução da AT&T em 1984 (que era então a maior do mundo est corporation, fornecendo serviço telefônico para cerca de 80 por cento dos Estados Unidos telefones), e a Lei de Telecomunicações de 1996 que revisou a regulamentação para fomentar a competição.

No outro extremo estão os países em que o governo nacional tem um monopólio completo de todas as comunicações, incluindo correio, telégrafo, telefone e, frequentemente, rádio e televisão. Grande parte do mundo se enquadra nesta categoria. Em alguns casos, a autoridade de telecomunicações é uma empresa nacionalizada, e em outros, é simplesmente um ramo do governo, geralmente conhecido como **PTT** (**Post, Administração de telégrafo e telefone**). Em todo o mundo, a tendência é liberalização e competição e longe do monopólio governamental. Mais europeus países já privatizaram (parcialmente) seus PTTs, mas em outros lugares o processo é ainda lentamente ganhando força.

Com todos esses diferentes fornecedores de serviços, há claramente uma necessidade de fornecer compatibilidade em escala mundial para garantir que as pessoas (e computadores) em um país pode chamar seus homólogos em outro. Na verdade, essa necessidade existia por muito tempo. Em 1865, representantes de muitos governos europeus se reuniram para formar o predecessor do atual **ITU** (**International Telecommunication União**). Seu trabalho era padronizar as telecomunicações internacionais, que em aqueles dias significavam telegrafia. Mesmo assim, ficou claro que se metade dos países usassem O código Morse e a outra metade usava algum outro código, haveria um problema lem. Quando o telefone foi colocado em serviço internacional, a ITU assumiu o trabalho de padronizar a telefonia (pronuncia-se te-LEF-on) também. Em 1947, ITU tornou-se uma agência das Nações Unidas.

A ITU tem cerca de 200 membros governamentais, incluindo quase todos os membros da as Nações Unidas. Como os Estados Unidos não têm um PTT, outra pessoa teve que representá-lo na UIT. Essa tarefa coube ao Departamento de Estado, provavelmente no motivos de que a ITU tinha a ver com países estrangeiros, o Departamento de Estado cialty. A ITU também tem mais de 700 membros setoriais e associados. Eles incluem companhias telefônicas (por exemplo, AT&T, Vodafone, Sprint), fabricantes de equipamentos de telecomunicações fabricantes (por exemplo, Cisco, Nokia, Nortel), fornecedores de computadores (por exemplo, Microsoft, Agilent, Toshiba), fabricantes de chips (por exemplo, Intel, Motorola, TI) e outras empresas interessadas empresas (por exemplo, Boeing, CBS, VeriSign).

A ITU tem três setores principais. Vamos nos concentrar principalmente no **ITU-T** , o Telecommunications setor de padronização de comunicações, que se preocupa com telefone e dados sistemas de comunicação. Antes de 1993, esse setor era denominado **CCITT** , que é um acrônimo de seu nome francês, Comité Consultatif International Télégraphique et Téléphonique. **ITU-R** , o Setor de Radiocomunicações, está preocupado com

Página 102

78
INTRODUÇÃO
INDIVÍDUO. 1

coordenar o uso de frequências de rádio em todo o mundo por grupos de interesse concorrentes. O outro setor é o ITU-D, o Setor de Desenvolvimento. Promove o desenvolvimento de tecnologias de informação e comunicação para estreitar a "divisão digital" entre países com acesso efetivo às tecnologias de informação e países tenta com acesso limitado.

A tarefa do ITU-T é fazer recomendações técnicas sobre telefone, telegráfico e interfaces de comunicação de dados. Muitas vezes se tornam internacionalmente padrões reconhecidos, embora tecnicamente as recomendações sejam apenas sugestões que os governos podem adotar ou ignorar, como quiserem (porque os governos são como meninos de 13 anos - eles não gostam de receber ordens). No prática, um país que deseja adotar um padrão telefônico diferente daquele usado pelo resto do mundo é livre para fazê-lo, mas ao preço de se isolar de todos os outros. Isso pode funcionar para a Coreia do Norte, mas em outros lugares seria um problema real.

O verdadeiro trabalho do ITU-T é feito em seus grupos de estudo. Existem atualmente 10 Grupos de estudo, muitas vezes com até 400 pessoas, que cobrem tópicos que vão desde telefaturamento de telefone para serviços multimídia para segurança. SG 15, por exemplo, padroniza as tecnologias DSL popularmente usadas para se conectar à Internet. Para fazer possível fazer qualquer coisa, os grupos de estudo são divididos em Partes, que por sua vez são divididas em equipes de especialistas, que por sua vez são divididas em grupos ad hoc. Uma vez burocracia, sempre burocracia.

Apesar de tudo isso, o ITU-T realmente realiza as coisas. Desde o seu início, produziu mais de 3.000 recomendações, muitas das quais são amplamente utilizadas na prática. Por exemplo, a Recomendação H.264 (também um padrão ISO conhecido como MPEG-4 AVC) é amplamente utilizado para compressão de vídeo e chave pública X.509 os certificados são usados para navegação segura na Web e e-mail assinado digitalmente. Com a conclusão do campo das telecomunicações, a transição começou no 1980 de ser totalmente nacional para ser totalmente global, os padrões se tornarão cada vez mais importante, e mais e mais organizações vão querer se tornar envolvidos em defini-los. Para obter mais informações sobre a ITU, consulte Irmer (1994).

1.6.2 Quem é quem no mundo dos padrões internacionais

Os padrões internacionais são produzidos e publicados pela ISO (International Organização de Padrões

†), uma organização voluntária sem tratado fundada em 1946.

Seus membros são as organizações de padrões nacionais dos 157 países membros. Esses membros incluem ANSI (EUA), BSI (Grã-Bretanha), AFNOR (França), DIN (Alemanha) e 153 outros.

A ISO emite padrões em um número verdadeiramente vasto de assuntos, que vão desde nozes e parafusos (literalmente) para revestimentos de postes de telefone [para não mencionar grãos de cacau (ISO 2451), redes de pesca (ISO 1530), roupas íntimas femininas (ISO 4416) e alguns

† Para o purista, o verdadeiro nome da ISO é International Organization for Standardization.

outros assuntos que não se pode pensar que estão sujeitos à padronização]. Em questões de padrões de telecomunicações, ISO e ITU-T freqüentemente cooperam (ISO é um membro da ITU-T) para evitar a ironia de dois internautas oficiais e mutuamente incompatíveis padrões internacionais.

Mais de 17.000 padrões foram emitidos, incluindo os padrões OSI. ISO tem mais de 200 Comitês Técnicos (TCs), numerados na ordem de criação ção, cada uma tratando de um assunto específico. TC1 lida com as porcas e parafusos (standardização de passos de rosca de parafuso). JTC1 lida com tecnologia da informação, incluindo redes, computadores e software. É a primeira (e até agora única) junta

Comitê Técnico, criado em 1987 pela fusão do TC97 com as atividades da IEC, mais um órgão de padronização. Cada TC tem subcomitês (SCs) divididos em grupos de trabalho (GTs).

O verdadeiro trabalho é feito em grande parte nos WGs por mais de 100.000 voluntários em todo o mundo

Largo. Muitos desses "voluntários" são designados para trabalhar em questões ISO por seus empregadores, cujos produtos estão sendo padronizados. Outros são funcionários do governo profissionais que desejam que a maneira de fazer as coisas de seu país se torne internacional padrão. Os especialistas acadêmicos também atuam em muitos dos WGs.

O procedimento usado pela ISO para a adoção de padrões foi projetado para alcançar um consenso o mais amplo possível. O processo começa quando um dos organizações de padrões nacionais sentem a necessidade de um padrão internacional em alguma área. Um grupo de trabalho é então formado para chegar a um **CD (Comitê Rascunho)**. O CD é então distribuído a todos os órgãos membros, que têm 6 meses para criticá-lo. Se uma maioria substancial aprovar, um documento revisado, denominado **DIS (Rascunho da Norma Internacional)** é produzido e distribuído para comentários e votação. Com base nos resultados desta rodada, o texto final do **IS (International Padrão)** é preparado, aprovado e publicado. Em áreas de grande controvérsia, um O CD ou DIS pode ter que passar por várias versões antes de adquirir o suficiente votos, e todo o processo pode levar anos.

NIST (Instituto Nacional de Padrões e Tecnologia) faz parte dos EUA Departamento de Comércio. Costumava ser chamado de National Bureau of Standards. Ele emite padrões que são obrigatórios para compras feitas pelo governo dos EUA mentos, exceto para os do Departamento de Defesa, que define sua própria norma dardos.

Outro jogador importante no mundo dos padrões é o **IEEE (Institute of Electrical e Engenheiros Eletrônicos)**, a maior organização profissional do mundo. Além de publicar dezenas de periódicos e realizar centenas de conferências a cada ano, o IEEE tem um grupo de padronização que desenvolve padrões na área de engenharia elétrica e computação. O comitê 802 do IEEE padronizou muitos tipos de LANs. Estudaremos alguns de seus resultados posteriormente neste livro. O AC- O trabalho real é realizado por uma coleção de grupos de trabalho, listados na Figura 1.38. A taxa de sucesso dos vários 802 grupos de trabalho tem sido baixa; tendo um 802.x número não é garantia de sucesso. Ainda assim, o impacto das histórias de sucesso (especialmente especialmente 802.3 e 802.11) na indústria e no mundo tem sido enorme.

Página 104

80

INTRODUÇÃO

INDIVÍDUO. 1

Número

Tema

802.1

Visão geral e arquitetura de LANs

802.2 ↓

Controle de link lógico

802.3 *

Ethernet

802.4 ↓

Barramento de token (foi usado brevemente em fábricas)

802.5

Token ring (a entrada da IBM no mundo LAN)

802.6 ↓

Barramento duplo de fila dupla (rede de área metropolitana anterior)

802.7 ↓

Grupo de assessoria técnica em tecnologias de banda larga

802.8 †

Grupo de assessoria técnica em tecnologias de fibra óptica

802.9 ↓

LANs isócronas (para aplicativos em tempo real)

802.10 ↓

LANs virtuais e segurança
 802.11 *
 LANs sem fio (WiFi)
 802.12 ↓
 Prioridade de demanda (AnyLAN da Hewlett-Packard)
 802.13
 Número azarado; ninguém queria isso
 802.14 ↓
 Modems a cabo (extinto: um consórcio da indústria chegou primeiro)
 802.15 *
 Redes de área pessoal (Bluetooth, Zigbee)
 802.16 *
 Banda larga sem fio (WiMAX)
 802.17
 Anel de pacote resiliente
 802.18
 Grupo de assessoria técnica sobre questões regulatórias de rádio
 802.19
 Grupo de assessoria técnica sobre a coexistência de todos esses padrões
 802.20
 Banda larga móvel sem fio (semelhante a 802.16e)
 802.21
 Transferência independente de mídia (para tecnologias de roaming)
 802.22
 Rede de área regional sem fio

Figura 1-38. Os 802 grupos de trabalho. Os importantes são marcados com *.

Os marcados com ↓ estão hibernando. Aquele marcado com † desistiu e se desfez.

1.6.3 Quem é quem no mundo dos padrões da Internet

A Internet mundial tem seus próprios mecanismos de padronização, muito diferente daqueles de ITU-T e ISO. A diferença pode ser resumida de forma crua dizendo que as pessoas que vêm às reuniões de padronização da ITU ou ISO usam ternos, enquanto as pessoas que vêm às reuniões de padronização da Internet usam jeans (exceto quando se encontram em San Diego, quando usam shorts e camisetas).

As reuniões da ITU-T e ISO são ocupadas por funcionários da empresa e do governo funcionários públicos para os quais a padronização é sua função. Eles consideram a padronização como uma coisa boa e devotar suas vidas a isso. O pessoal da Internet, por outro lado, prefere a anarquia por uma questão de princípio. No entanto, com centenas de milhões de

SEC. 1,6
PADRONIZAÇÃO DA REDE

81

pessoas fazendo suas próprias coisas, pouca comunicação pode ocorrer. Assim, padrões, por mais lamentáveis que sejam, às vezes são necessários. Neste contexto, David Clark da O MIT uma vez fez uma observação agora famosa sobre a padronização da Internet que consiste de " consenso aproximado e código em execução. "

Quando a ARPANET foi criada, o DoD criou um comitê informal para supervisionar. Em 1983, o comitê foi renomeado para **IAB (Internet Activities Board)** e recebeu uma missão mais ampla, ou seja, manter os pesquisadores envolvidos com a ARPANET e a Internet apontavam mais ou menos na mesma direção, uma atividade não muito diferente de pastorear gatos. O significado da sigla " IAB " foi posteriormente alterado para **Internet Architecture Board** .

Cada um dos aproximadamente dez membros do IAB liderou uma força-tarefa em alguma questão de importância. O IAB se reunia várias vezes por ano para discutir os resultados e para dar feedback ao DoD e NSF, que estavam fornecendo a maior parte do financiamento neste momento. Quando um padrão era necessário (por exemplo, um novo algoritmo de roteamento),

os membros do IAB discutiriam e, em seguida, anunciariam a mudança para que o graduado Até os alunos que eram o coração do esforço de software poderiam implementá-lo. Comunicação foi feita por uma série de relatórios técnicos chamados **RFCs (Request For**

Comentários). RFCs são armazenados online e podem ser obtidos por qualquer pessoa interessada em eles de www.ietf.org/rfc. Eles são numerados em ordem cronológica de criação ção. Já existem mais de 5.000. Faremos referência a muitos RFCs neste livro. Em 1989, a Internet havia crescido tanto que esse estilo altamente informal não funcionou mais. Muitos fornecedores até então ofereciam produtos TCP / IP e não queriam mudá-los só porque dez pesquisadores tiveram uma ideia melhor. No verão de 1989, o IAB foi reorganizado novamente. Os pesquisadores foram transferidos para o **IRTF (Internet Research Task Force)**, que se tornou subsidiária do IAB, junto com o **IETF (Internet Engineering Task Force)**. O IAB foi repovoado com pessoas que representam uma gama mais ampla de organizações do que apenas comunidade de pesquisa. Era inicialmente um grupo que se autoperpetua, com membros servindo por um mandato de 2 anos e os novos membros indicados pelos antigos. Mais tarde, foi criada a **Internet Society**, composta por pessoas interessadas na Internet. A Internet Society é, portanto, em certo sentido, comparável à ACM ou IEEE. Isto é governado por curadores eleitos que indicam os membros do IAB. A ideia dessa divisão era fazer com que o IRTF se concentrasse em pesquisas de longo prazo enquanto o IETF lidou com questões de engenharia de curto prazo. O IETF foi dividido em grupos de trabalho, cada um com um problema específico para resolver. Os presidentes de esses grupos de trabalho se reuniram inicialmente como um comitê diretor para dirigir o engenheiro esforço ing. Os tópicos do grupo de trabalho incluem novos aplicativos, informações do usuário, Integração OSI, roteamento e endereçamento, segurança, gerenciamento de rede e padrão dardos. Eventualmente, tantos grupos de trabalho foram formados (mais de 70) que eles foram agrupados em áreas e os presidentes de área se reuniram como o comitê de direção. Além disso, um processo de padronização mais formal foi adotado, padronizado após ISOs. Para se tornar um **padrão proposto**, a ideia básica deve ser explicada em uma RFC e ter interesse suficiente na comunidade para justificar consideração.

Página 106

82

INTRODUÇÃO
INDIVÍDUO. 1

Para avançar para o estágio de **Rascunho do Padrão**, uma implementação funcional deve ter sido rigorosamente testado por pelo menos dois sites independentes por pelo menos 4 meses. E se o IAB está convencido de que a ideia é sólida e o software funciona, pode declarar o RFC para ser um **padrão da Internet**. Alguns padrões da Internet tornaram-se Padrões DoD (MIL-STD), tornando-os obrigatórios para fornecedores DoD. Para os padrões da Web, o **World Wide Web Consortium (W3C)** desenvolve ferramentas e diretrizes para facilitar o crescimento da web a longo prazo. É uma indústria tente consórcio liderado por Tim Berners-Lee e criado em 1994 como a Web realmente começou a decolar. W3C agora tem mais de 300 membros de todo o mundo e produziu mais de 100 recomendações W3C, pois seus padrões são chamado, cobrindo tópicos como HTML e privacidade na Web.

1.7 UNIDADES MÉTRICAS

Para evitar qualquer confusão, vale a pena afirmar explicitamente que neste livro, como em ciência da computação em geral, unidades métricas são usadas em vez do inglês tradicional unidades (o sistema furlong-stone-quinzena). Os principais prefixos métricos são listados na Figura 1-39. Os prefixos são normalmente abreviados pelas primeiras letras, com o unidades maiores que 1 em maiúsculas (KB, MB, etc.). Uma exceção (por motivos históricos filhos) é kbps para kilobits / s. Assim, uma linha de comunicação de 1 Mbps transmite 10^6 bits / seg e um relógio de 100 psec (ou 100×10^{-12} s) tiqueteia a cada 10^{-10} segundos. Desde mili

e micro começam com a letra " m ", uma escolha teve que ser feita. Normalmente, " m " é usado para mili e " μ " (a letra grega mu) é usado para micro.

Exp.
Explícito
Prefixo

Exp.	
Explícito	
Prefixo	
10^{-3}	
0,001	
mili	
10^3	
1.000	
Quilo	
10^{-6}	
0,000001	
micro	
10^6	
1.000.000	
Mega	
10^{-9}	
0,000000001	
nano	
10^9	
1.000.000.000	
Giga	
10^{-12}	
0,000000000001	
pico	
10^{12}	
1.000.000.000.000	
Tera	
10^{-15}	
0,000000000000001	
femto	
10^{15}	
1.000.000.000.000.000	
Peta	
10^{-18}	
0,0000000000000000001	
atto	
10^{18}	
1.000.000.000.000.000.000	
Exa	
10^{-21}	
0,00000000000000000000001	
zepto	
10^{21}	
1.000.000.000.000.000.000.000	
Zetta	
10^{-24}	
0,00000000000000000000000000001	
yocto	
10^{24}	
1.000.000.000.000.000.000.000.000	
Yotta	

Figura 1-39. Os principais prefixos métricos.

Também vale a pena apontar que, para medir memória, disco, arquivo e dados, tamanhos de base, na prática comum da indústria, as unidades têm médias ligeiramente diferentes ings. Lá, quilo significa 2^{10} (1024) em vez de 10^3 (1000) porque as memórias são sempre uma potência de dois. Assim, uma memória de 1 KB contém 1.024 bytes, não 1000 bytes. Observe também o " B " maiúsculo nesse uso para significar " bytes " (unidades de oito

Página 107

SEC. 1,7
UNIDADES MÉTRICAS

83

bits), em vez de " b " minúsculo que significa " bits ". Da mesma forma, uma memória de 1 MB contém 2^{20} (1.048.576) bytes, uma memória de 1 GB contém 2^{30} (1.073.741.824) bytes, e um banco de dados de 1 TB contém 2^{40} (1.099.511.627.776) bytes. No entanto, um A linha de comunicação de 1 kbps transmite 1000 bits por segundo e uma LAN de 10 Mbps funciona a 10.000.000 bits / s porque essas velocidades não são potências de dois. Infelizmente, muitas pessoas tendem a misturar esses dois sistemas, especialmente para tamanhos de disco.

Para evitar ambigüidade, neste livro, usaremos os símbolos KB, MB, GB e TB para 2^{10} , 2^{20} , 2^{30} e 2^{40} bytes, respectivamente, e os símbolos kbps, Mbps, Gbps, e Tbps para 10^3 , 10^6 , 10^9 e 10^{12} bits / s, respectivamente.

1.8 ESBOÇO DO RESTO DO LIVRO

Este livro discute os princípios e a prática das redes de computadores.

A maioria dos capítulos começa com uma discussão dos princípios relevantes, seguido por um vários exemplos que ilustram esses princípios. Esses exemplos são geralmente retirado da Internet e de redes sem fio, como a rede de telefonia móvel uma vez que são importantes e muito diferentes. Outros exemplos serão dados onde for relevante.

O livro está estruturado de acordo com o modelo híbrido da Figura 1.23. Iniciando com cap. 2, começamos a trabalhar nosso caminho para cima na hierarquia de protocolo começando em

o fundo. Fornecemos alguns antecedentes no campo da comunicação de dados que abrange sistemas de transmissão com e sem fio. Este material é preocupante sobre como fornecer informações por canais físicos, embora cubramos apenas os aspectos arquitetônicos em vez dos de hardware. Vários exemplos de camada cal, como a rede telefônica pública comutada, o telefone móvel rede, e a rede de televisão a cabo também são discutidos.

Os capítulos 3 e 4 discutem a camada de enlace em duas partes. Indivíduo. 3 olhares para o problema de como enviar pacotes através de um link, incluindo detecção de erro e correção. Vemos DSL (usado para acesso à Internet de banda larga em linhas telefônicas) como um exemplo do mundo real de um protocolo de link de dados.

No cap. 4, examinamos a subcamada de acesso ao meio. Esta é a parte do camada de enlace de dados que trata de como compartilhar um canal entre vários puters. Os exemplos que examinamos incluem wireless, como 802.11 e RFID, e LANs com fio, como Ethernet clássica. Switches de camada de link que conectam LANs, como Ethernet comutada, também são discutidos aqui.

O Capítulo 5 trata da camada de rede, especialmente do roteamento. Muitos algoritmos de roteamento

ritmos, estáticos e dinâmicos, são cobertos. Mesmo com bons algoritmos de roteamento, porém, se for oferecido mais tráfego do que a rede pode suportar, alguns pacotes ser adiado ou descartado. Discutimos esta questão de como evitar o congestionamento para como garantir uma certa qualidade de serviço. Conectando rede heterogênea-trabalha para formar internetworks também leva a vários problemas que são discutidos aqui. A camada de rede na Internet recebe ampla cobertura.

Página 108

84

INTRODUÇÃO INDIVÍDUO. 1

O Capítulo 6 trata da camada de transporte. Muito da ênfase está na conexão protocolos orientados por ção e confiabilidade, uma vez que muitos aplicativos precisam deles. Ambos

Os protocolos de transporte da Internet, UDP e TCP, são abordados em detalhes, assim como seus questões de desempenho.

O Capítulo 7 trata da camada de aplicação, seus protocolos e suas aplicações.

O primeiro tópico é DNS, que é a lista telefônica da Internet. Em seguida, vem o e-mail, incluindo uma discussão de seus protocolos. Em seguida, passamos para a Web, com discussões personalizadas de conteúdo estático e dinâmico e o que acontece no cliente e os lados do servidor. Seguimos isso com uma olhada na multimídia em rede, incluindo streaming de áudio e vídeo. Finalmente, discutimos as redes de entrega de conteúdo, incluindo tecnologia peer-to-peer.

O Capítulo 8 é sobre segurança de rede. Este tópico tem aspectos que se relacionam a todos camadas, por isso é mais fácil tratá-lo depois que todas as camadas foram completamente explicadas-

ed. O capítulo começa com uma introdução à criptografia. Mais tarde, mostra como a criptografia pode ser usada para proteger a comunicação, e-mail e a web. o capítulo termina com uma discussão de algumas áreas em que a segurança colide com privacidade, liberdade de expressão, censura e outras questões sociais.

O Capítulo 9 contém uma lista anotada de leituras sugeridas organizadas por cap-

ter. Destina-se a ajudar os leitores que desejam continuar seus estudos de rede ainda mais. O capítulo também contém uma bibliografia alfabética de todas as referências citadas neste livro.

O site dos autores na Pearson:

<http://www.pearsonhighered.com/tanenbaum>

tem uma página com links para muitos tutoriais, perguntas frequentes, empresas, consórcios da indústria, pro

organizações profissionais, organizações de padrões, tecnologias, documentos e muito mais.

1.9 RESUMO

As redes de computadores têm muitos usos, tanto para empresas como para indivíduos, em casa e em trânsito. As empresas usam redes de computadores para compartilhar informações corporativas, normalmente usando o modelo cliente-servidor com desktops de funcionários atuando como clientes acessando servidores poderosos na máquina quarto. Para os indivíduos, as redes oferecem acesso a uma variedade de informações e recursos de entretenimento, bem como uma forma de compra e venda de produtos e serviços. As pessoas costumam acessar a Internet por meio de seus provedores de telefone ou cabo em casa, embora cada vez mais o acesso sem fio seja usado para laptops e telefones. Tecnologia avanços estão permitindo novos tipos de aplicativos e redes móveis com computadores embutidos em aparelhos e outros dispositivos de consumo. Os mesmos avanços levantam questões sociais, como questões de privacidade.

A grosso modo, as redes podem ser divididas em LANs, MANs, WANs e internetworks. As LANs típicas cobrem um edifício e operam em alta velocidade. MANs

Página 109

SEC. 1,9
RESUMO

85

geralmente cobrem uma cidade. Um exemplo é o sistema de televisão a cabo, que agora é usado por muitas pessoas para acessar a Internet. As WANs podem cobrir um país ou continente. Algumas das tecnologias usadas para construir essas redes são ponto a ponto (por exemplo, um cabo) enquanto outros são transmitidos (por exemplo, sem fio). As redes podem ser interconectadas com roteadores para formar internetworks, das quais a Internet é a maior e melhor exemplo conhecido. Redes sem fio, por exemplo, 802.11 LANs e 3G móveis telefonia, também estão se tornando extremamente populares.

O software de rede é construído em torno de protocolos, que são regras pelas quais processos se comunicam. A maioria das redes oferece suporte a hierarquias de protocolo, com cada camada

fornecer serviços para a camada acima dela e isolá-los dos detalhes do protocolos usados nas camadas inferiores. Pilhas de protocolo são normalmente baseadas em no modelo OSI ou no modelo TCP / IP. Ambos têm link, rede, transporte e camadas de aplicação, mas elas diferem nas outras camadas. Problemas de design incluem confiabilidade, alocação de recursos, crescimento, segurança e muito mais. Muito deste livro lida com protocolos e seu design.

As redes fornecem vários serviços aos seus usuários. Esses serviços podem variar desde a entrega de pacotes de melhores esforços sem conexão até a garantia orientada à conexão entrega teed. Em algumas redes, o serviço sem conexão é fornecido em uma camada e o serviço orientado à conexão é fornecido na camada acima dele.

Redes conhecidas incluem a Internet, a rede de telefonia móvel 3G, e LANs 802.11. A Internet evoluiu da ARPANET, para a qual outras redes obras foram adicionadas para formar uma internetwork. A Internet atual é na verdade um coleção de muitos milhares de redes que usam a pilha de protocolo TCP / IP. A rede de telefonia móvel 3G fornece acesso móvel e sem fio à Internet a velocidades de vários Mbps e, é claro, também transporta chamadas de voz. Sem fio LANs baseadas no padrão IEEE 802.11 são implantadas em muitas casas e cafés e pode fornecer conectividade a taxas superiores a 100 Mbps. Novos tipos de rede trabalhos estão surgindo também, como redes de sensores incorporados e redes baseadas na tecnologia RFID.

Permitir que vários computadores conversem entre si requer uma grande quantidade de padronização, tanto no hardware quanto no software. Organizações como ITU-T, ISO, IEEE e IAB gerenciam diferentes partes do processo de padronização.

PROBLEMAS

1. Imagine que você treinou seu São Bernardo, Bernie, para transportar uma caixa de três 8 mm fitas em vez de uma garrafa de conhaque. (Quando o disco fica cheio, você considera que um emergência.) Cada uma dessas fitas contém 7 gigabytes. O cachorro pode viajar para o seu lado, onde quer que você esteja, a 18 km / hora. Para qual intervalo de distâncias Bernie tem um taxa de dados mais alta do que uma linha de transmissão cuja taxa de dados (excluindo sobrecarga) é 150 Mbps? Como sua resposta muda se (i) a velocidade de Bernie dobrar; (ii) cada fita a capacidade é duplicada; (iii) a taxa de dados da linha de transmissão é dobrada.

Página 110

86

INTRODUÇÃO

INDIVÍDUO. 1

2. Uma alternativa para uma LAN é simplesmente um grande sistema de compartilhamento de tempo com terminais para todos

Comercial. Dê duas vantagens de um sistema cliente-servidor usando uma LAN.

3. O desempenho de um sistema cliente-servidor é fortemente influenciado por duas grandes redes características de trabalho: a largura de banda da rede (ou seja, quantos bits / s pode transporte) e a latência (ou seja, quantos segundos leva para o primeiro bit chegar do cliente para o servidor). Dê um exemplo de rede que exibe banda alta largura, mas também alta latência. Em seguida, dê um exemplo de um que tenha baixa largura de banda e baixa latência.

4. Além da largura de banda e latência, que outro parâmetro é necessário para dar um bom caráter terização da qualidade do serviço oferecido por uma rede utilizada para (i) voz digitalizada tráfego? (ii) tráfego de vídeo? (iii) tráfego de transações financeiras?

5. Um fator no atraso de um sistema de comutação de pacotes de armazenamento e encaminhamento é quanto tempo ele

leva para armazenar e encaminhar um pacote por meio de um switch. Se o tempo de comutação for 10 µseg, é este provavelmente será um fator importante na resposta de um sistema cliente-servidor onde o cliente ent está em Nova York e o servidor está na Califórnia? Suponha que a velocidade de propagação em cobre e fibra a 2/3 da velocidade da luz no vácuo.

6. Um sistema cliente-servidor usa uma rede de satélite, com o satélite a uma altura de 40.000 km. Qual é o melhor caso de atraso em resposta a uma solicitação?

7. No futuro, quando todos tiverem um terminal doméstico conectado a uma rede de computadores, referendos públicos instantâneos sobre legislação importante pendente tornar-se-ão possíveis.

Em última análise, as legislaturas existentes poderiam ser eliminadas, para deixar a vontade do povo ser expressa diretamente. Os aspectos positivos de tal democracia direta são bastante óbvios; discuta alguns dos aspectos negativos.

8. Cinco roteadores devem ser conectados em uma sub-rede ponto a ponto. Entre cada par de roteadores, os projetistas podem colocar uma linha de alta velocidade, uma linha de média velocidade, uma linha de baixa velocidade

linha ou nenhuma linha. Se levar 100 ms do tempo do computador para gerar e inspecionar cada topologia, quanto tempo levará para inspecionar todos eles?

9. Uma desvantagem de uma sub-rede de broadcast é a capacidade desperdiçada quando vários hosts em tentação de acessar o canal ao mesmo tempo. Como um exemplo simplista, suponha que o tempo é dividido em slots discretos, com cada um dos n hosts tentando usar o canal nel com probabilidade p durante cada slot. Que fração dos slots será desperdiçada devido às colisões?

10. Quais são as duas razões para usar protocolos em camadas? Qual é uma possível desvantagem de usar protocolos em camadas?

11. O presidente da Specialty Paint Corp. tem a ideia de trabalhar com uma cerveja local cervejeiro para produzir uma lata de cerveja invisível (como uma medida anti-lixo). O presidente conta seu departamento jurídico para investigar isso e, por sua vez, pedir ajuda à engenharia. Como são-sultado, o engenheiro-chefe liga para seu homólogo na cervejaria para discutir as questões técnicas aspectos do projeto. Os engenheiros, então, se reportam ao seu respectivo departamento jurídico mentos, que então se comunicam por telefone para acertar os aspectos jurídicos. Finalmente, os dois presidentes corporativos discutem o lado financeiro do negócio. Que princípio de um multi-protocolo de camada no sentido do modelo OSI faz este mecanismo de comunicação violar?

Página 111

INDIVÍDUO. 1

PROBLEMAS

87

12. Duas redes, cada uma, fornecem serviço orientado a conexão confiável. Um deles oferece um fluxo de bytes confiável e o outro oferece um fluxo de mensagens confiável. Estes são idênticos? Em caso afirmativo, por que a distinção é feita? Se não, dê um exemplo de como eles diferem.
13. O que significa "negociação" quando se discute protocolos de rede? Dê um exemplo.
14. Na Figura 1-19, um serviço é mostrado. Existem outros serviços implícitos nesta figura? Se então, Onde? Se não, porque não?
15. Em algumas redes, a camada de enlace de dados lida com erros de transmissão, solicitando que quadros danificados sejam retransmitidos. Se a probabilidade de um quadro ser danificado é p , qual é o número médio de transmissões necessárias para enviar um quadro? Assuma isso reconhecimentos nunca são perdidos.
16. Um sistema tem uma hierarquia de protocolo de n camadas. Os aplicativos geram mensagens de comprimento Bytes M . Em cada uma das camadas, um cabeçalho de h - byte é adicionado. Que fração da rede largura de banda de trabalho está cheia de cabeçalhos?
17. Qual é a principal diferença entre TCP e UDP?
18. A sub-rede da Figura 1.25 (b) foi projetada para resistir a uma guerra nuclear. Quantos seriam necessários bombas para partitionar os nós em dois conjuntos desconectados? Assuma isso qualquer bomba apaga um nó e todos os links conectados a ele.
19. A Internet praticamente dobra de tamanho a cada 18 meses. Embora ninguém realmente saiba com certeza, uma estimativa coloca o número de hosts nele em 600 milhões em 2009. Use esses dados para calcular o número esperado de hosts da Internet no ano de 2018. Acredita nisso? Explique por que ou por que não.
20. Quando um arquivo é transferido entre dois computadores, duas estratégias de confirmação é possível. No primeiro, o arquivo é dividido em pacotes, que são individuais reconhecido pelo receptor, mas a transferência do arquivo como um todo não é saliente. No segundo, os pacotes não são reconhecidos individualmente, mas o arquivo de pneu é reconhecido quando chega. Discuta essas duas abordagens.
21. As operadoras de rede de telefonia móvel precisam saber onde estão os telefones celulares de seus assinantes (daí seus usuários) são localizados. Explique por que isso é ruim para os usuários. Agora dê razões por que isso é bom para os usuários.
22. Quanto tempo tinha um bit no padrão 802.3 original em metros? Use uma velocidade de transmissão de 10 Mbps e assumir que a velocidade de propagação no cabo coaxial é $2/3$ da velocidade da luz em vácuo.
23. Uma imagem tem 1600×1200 pixels com 3 bytes / pixel. Suponha que a imagem seja descompactado. Quanto tempo leva para transmiti-lo por um canal de modem de 56 kbps? Por meio de um modem a cabo de 1 Mbps? Em uma Ethernet de 10 Mbps? Mais de Ethernet de 100 Mbps? Por Ethernet gigabit?
24. Ethernet e redes sem fio têm algumas semelhanças e algumas diferenças. 1 propriedade da Ethernet é que apenas um quadro por vez pode ser transmitido em uma Ethernet. O 802.11 compartilha essa propriedade com a Ethernet? Discuta sua resposta.
25. Liste duas vantagens e duas desvantagens de ter padrões internacionais para protocolos de trabalho.

Página 112

88

INTRODUÇÃO

INDIVÍDUO. 1

26. Quando um sistema tem uma parte permanente e uma parte removível (como uma unidade de CD-ROM e o CD-ROM), é importante que o sistema seja padronizado, para que diferentes empresas podem fazer as partes permanentes e removíveis e tudo ainda trabalha junto. Dê três exemplos fora da indústria de computadores, onde tais existem padrões nacionais. Agora dê três áreas fora da indústria de informática onde eles não existem.
27. Suponha que os algoritmos usados para implementar as operações na camada k sejam alterados. Como isso afeta as operações nas camadas $k - 1$ e $k + 1$?
28. Suponha que haja uma mudança no serviço (conjunto de operações) fornecido pela camada k . Como isso afeta os serviços nas camadas $k - 1$ e $k + 1$?
29. Forneça uma lista de razões pelas quais o tempo de resposta de um cliente pode ser maior do que o atraso na melhor das hipóteses.
30. Quais são as desvantagens de usar células pequenas e de comprimento fixo em ATM?
31. Faça uma lista das atividades que você realiza todos os dias nas quais são usadas redes de computadores. Como sua vida seria alterada se essas redes fossem desligadas repentinamente?
32. Descubra quais redes são usadas em sua escola ou local de trabalho. Descreva a rede

tipos de trabalho, topologias e métodos de comutação usados lá.

33. O programa *ping* permite que você envie um pacote de teste para um determinado local e veja como o tempo que leva para ir e voltar. Tente usar o *ping* para ver quanto tempo leva para sair sua localização para vários locais conhecidos. A partir desses dados, trace o trânsito unilateral tempo na Internet em função da distância. É melhor usar universidades, já que a localização de seus servidores é conhecida com muita precisão. Por exemplo, *berkeley.edu* está em Berkeley, Califórnia; *mit.edu* está em Cambridge, Massachusetts; *vu.nl* fica em Amsterdã; Os Países Baixos; *www.usyd.edu.au* está localizado em Sydney, Austrália; e *www.uct.ac.za* está em Cidade do Cabo, África do Sul.

34. Acesse o site da IETF, www.ietf.org, para ver o que estão fazendo. Escolha um projeto para você curta e escreva um relatório de meia página sobre o problema e a solução proposta.

35. A Internet é composta por um grande número de redes. Seu arranjo determina a topologia da Internet. Uma quantidade considerável de informações sobre a Internet a topologia está disponível online. Use um mecanismo de pesquisa para saber mais sobre a Internet topologia e escrever um breve relatório resumindo suas descobertas.

36. Pesquise na Internet para descobrir alguns dos pontos de peering importantes usados para roteamento pacotes na Internet atualmente.

37. Escreva um programa que implemente o fluxo de mensagens da camada superior para a inferior do modelo de protocolo de 7 camadas. Seu programa deve incluir uma função de protocolo separada para cada camada. Os cabeçalhos de protocolo são sequenciais de até 64 caracteres. Cada protocolo função tem dois parâmetros: uma mensagem passada do protocolo da camada superior (um char buffer) e o tamanho da mensagem. Esta função anexa seu cabeçalho na frente da mensagem, imprime a nova mensagem na saída padrão e, em seguida, invoca o protocolo função do protocolo da camada inferior. A entrada do programa é uma mensagem do aplicativo (uma sequência de 80 caracteres ou menos).

Página 113

2

A CAMADA FÍSICA

Neste capítulo, veremos a camada mais baixa em nosso modelo de protocolo, o camada física. Ele define a elétrica, o tempo e outras interfaces pelas quais os bits são enviados como sinais pelos canais. A camada física é a base sobre a qual a rede é construída. As propriedades de diferentes tipos de canais físicos determinam extrair o desempenho (por exemplo, rendimento, latência e taxa de erro) para que seja um bom lugar para começar nossa jornada em networkland.

Começaremos com uma análise teórica da transmissão de dados, apenas para descobrir que a Mãe (Pai?) Natureza coloca alguns limites sobre o que pode ser enviado através de um canal. Em seguida, cobriremos três tipos de meios de transmissão: guiados (cobre fio e fibra óptica), sem fio (rádio terrestre) e satélite. Cada um desses tecnologias tem propriedades diferentes que afetam o design e desempenho das redes que os utilizam. Este material fornecerá informações básicas sobre as principais tecnologias de transmissão usadas em redes modernas.

Em seguida, vem a modulação digital, que trata de como os sinais analógicos são convertidos em bits digitais e vice-versa. Depois disso, veremos a multiplexação esquemas, explorando como várias conversas podem ser colocadas na mesma transmissão meio de comunicação ao mesmo tempo, sem interferir um no outro.

Finalmente, veremos três exemplos de sistemas de comunicação usados em prática para redes de computadores de longa distância: o sistema de telefonia (fixa), o sistema de telefonia móvel e sistema de televisão a cabo. Cada um deles é importante na prática, dedicaremos uma boa quantidade de espaço a cada um.

89

Página 114

2.1 A BASE TEÓRICA PARA COMUNICAÇÃO DE DADOS

As informações podem ser transmitidas por fios variando algumas propriedades físicas como tensão ou corrente. Representando o valor desta tensão ou corrente como uma função de tempo de valor único, $f(t)$, podemos modelar o comportamento do sinal e analise matematicamente. Esta análise é o assunto das seções seguintes.

2.1.1 Análise de Fourier

No início do século 19, o matemático francês Jean-Baptiste Fourier provou que qualquer função periódica razoavelmente comportada, $g(t)$ com período T , pode ser construído como a soma de um número (possivelmente infinito) de senos e cossenos:

$$g(t) = c + \sum_{n=1}^{\infty} a_n \sin(2\pi nft) + \sum_{n=1}^{\infty} b_n \cos(2\pi nft) \quad (2-1)$$

onde $f = 1/T$ é a frequência fundamental, a_n e b_n são o seno e o cosseno amplitudes do n th **harmónicos** (termos), e c é uma constante. Essa decomposição é chamada de **série de Fourier**. Da série Fourier, a função pode ser reconstruído. Ou seja, se o período, T , é conhecido e as amplitudes são dadas, a origem

A função final do tempo pode ser encontrada executando as somas da Eq. (2-1).

Um sinal de dados que tem uma duração finita, o que todos eles têm, pode ser tratado apenas imaginando que ele repete todo o padrão indefinidamente (ou seja, o intervalo de T a $2T$ é o mesmo que de 0 a T , etc.).

As um_n amplitudes pode ser calculado para um determinado $g(t)$ através da multiplicação tanto lados da Eq. (2-1) pelo pecado ($2\pi KFT$) e depois integrando entre 0 e T . Desde a

$$\int_0^T \sin(2\pi kft) \sin(2\pi nft) dt = \begin{cases} T/2 & \text{para } k = n \\ 0 & \text{para } k \neq n \end{cases}$$

$T/2$ para $k = n$

0 para $k \neq n$

apenas um termo do somatório sobreviveu: a_n . O somatório b_n desaparece com completamente. Da mesma forma, multiplicando a Eq. (2-1) por $\cos(2\pi kft)$ e integração entre 0 e T , podemos derivar b_n . Apenas integrando ambos os lados da equação, uma vez que carrinhos, podemos encontrar c . Os resultados da realização dessas operações são os seguintes:

$$a_n = \frac{1}{T} \int_0^T g(t) \sin(2\pi nft) dt$$

$$b_n = \frac{1}{T} \int_0^T g(t) \cos(2\pi nft) dt$$

$$c =$$

$$g(t) = \int_0^T g(t) \cos(2\pi nft) dt$$

$$c = \frac{1}{T} \int_0^T g(t) dt$$

2.1.2 Sinais de largura de banda limitada

A relevância de tudo isso para a comunicação de dados é que os canais reais afetam sinais de frequência diferentes de forma diferente. Vamos considerar um exemplo específico: o transmissão do caractere ASCII "b" codificado em um byte de 8 bits. O padrão de bits que deve ser transmitido é 01100010. A parte esquerda da Figura 2-1 (a) mostra o

Página 115

SEC. 2.1 A BASE TEÓRICA PARA COMUNICAÇÃO DE DADOS

91

saída de tensão pelo computador transmissor. A análise de Fourier deste sinal produz os coeficientes:

$$a_n = \frac{1}{\pi n} [\cos(\pi n / 4) - \cos(3\pi n / 4) + \cos(6\pi n / 4) - \cos(7\pi n / 4)]$$

$$b_n = \frac{1}{\pi n} [\sin(3\pi n / 4) - \sin(\pi n / 4) + \sin(7\pi n / 4) - \sin(6\pi n / 4)]$$

$$c = 3/4$$

As amplitudes da raiz quadrada média, $\sqrt{a_n^2 + b_n^2}$

, pois os primeiros termos são mostrados em o lado direito da Figura 2-1 (a). Esses valores são interessantes porque seus quadrados são proporcionais à energia transmitida na frequência correspondente. Nenhuma instalação de transmissão pode transmitir sinais sem perder alguma potência no processo. Se todos os componentes de Fourier fossem igualmente diminuídos, o sinal resultante não seria reduzido em amplitude, mas não distorcido [ou seja, teria o mesmo bela forma quadrada como a Fig. 2-1 (a)]. Infelizmente, todas as instalações de transmissão diminuir diferentes componentes de Fourier em diferentes quantidades, introduzindo assim distorção. Normalmente, para um fio, as amplitudes são transmitidas principalmente sem diminuir de 0 até alguma frequência f_c [medido em ciclos / seg ou Hertz (Hz)], com todos as frequências acima desta frequência de corte atenuadas. A largura da frequência a faixa transmitida sem ser fortemente atenuada é chamada de **largura de banda**. No prática, o corte não é realmente nítido, então muitas vezes a largura de banda citada é de 0 a uma frequência com que a potência recebida caiu pela metade.

A largura de banda é uma propriedade física do meio de transmissão que depende, por exemplo, da construção, espessura e comprimento de um fio ou fibra. Os filtros são freqüentemente usados para limitar ainda mais a largura de banda de um sinal. 802.11 sem fio canais podem usar até cerca de 20 MHz, por exemplo, para rádios 802.11 filtrar a largura de banda do sinal para este tamanho. Como outro exemplo, tradicional (análogo) os canais de televisão ocupam 6 MHz cada, com fio ou sem fio. Esta filtragem permite que mais sinais compartilhem uma determinada região do espectro, o que melhora o efeito geral

eficiência do sistema. Isso significa que a faixa de frequência para alguns sinais não comece do zero, mas isso não importa. A largura de banda ainda é a largura do banda de frequências que são passadas, e as informações que podem ser transportadas depende apenas desta largura e não das frequências inicial e final. Sinais que vão de 0 até uma frequência máxima são chamados de sinais de **banda base**. Sinais que são deslocados para ocupar uma faixa mais alta de frequências, como é o caso de todos os fios menos transmissões, são chamadas de sinais de **banda passante**.

Agora, vamos considerar como o sinal da Figura 2.1 (a) ficaria se a largura de banda eram tão baixos que apenas as frequências mais baixas eram transmitidas [ou seja, se a função estavam sendo aproximados pelos primeiros termos da Eq. (2-1)]. Figura 2-1 (b) mostra o sinal que resulta de um canal que permite apenas o primeiro harmônico

Página 116

92

A CAMADA FÍSICA
INDIVÍDUO. 2

```

0
1
1
0
0
0
1
0
1
0
0
Tempo
T
1
0
1
0
1
0
1
0
1
0
Tempo
rms
amplitude
1
15
2 3 4 5 6 7
9 10 11 12 13 14
8
0,50
0,25
Número harmônico
1 harmônico
2 harmônicos
4 harmônicos
8 harmônicos
1
1 2
1 2 3 4
1 2 3 4 5 6 7 8
Número harmônico
(uma)
(b)
(c)
(d)
(e)
```

Figura 2-1. (a) Um sinal binário e suas amplitudes de Fourier quadradas médias.
(b) - (e) Aproximações sucessivas do sinal original.

Página 117

SEC. 2.1 A BASE TEÓRICA PARA COMUNICAÇÃO DE DADOS

93

(o fundamental, f) para passar. Da mesma forma, a Fig. 2-1 (c) - (e) mostra os espectros e funções reconstruídas para canais de maior largura de banda. Para transmissão digital seção, o objetivo é receber um sinal com fidelidade suficiente para reconstruir a seção sequência de bits que foi enviada. Já podemos fazer isso facilmente na Fig. 2-1 (e), então é desperdício de usar mais harmônicos para receber uma réplica mais precisa.

Dada uma taxa de bits de b bits / s, o tempo necessário para enviar os 8 bits em nosso exemplo 1 bit por vez é $8/b$ seg, então a frequência do primeiro harmônico deste sinal

nal é $b/8$ Hz. Uma linha telefônica comum, muitas vezes chamada **de linha de grau de voz**, tem

um

frequência de corte introduzida artificialmente logo acima de 3000 Hz. A presença deste restrição significa que o número do harmônico mais alto passado é aproximadamente $3000/(b/8)$, ou $24.000/b$ (o corte não é acentuado).

Para algumas taxas de dados, os números funcionam conforme mostrado na Figura 2-2. A partir destes

números, é claro que tentar enviar a 9600 bps por um telefone de qualidade linha irá transformar a Fig. 2-1 (a) em algo parecido com a Fig. 2-1 (c), tornando recepção precisa do fluxo de bits binário original difícil. Deveria ser óbvio que a taxas de dados muito superiores a 38,4 kbps, não há esperança para sinais *binários* nals, mesmo que a instalação de transmissão seja completamente silenciosa. Em outras palavras, limitar a largura de banda limita a taxa de dados, mesmo para canais perfeitos. No entanto, cod-esquemas de manipulação que fazem uso de vários níveis de tensão existem e podem alcançar mais taxas de dados. Discutiremos isso mais tarde neste capítulo.

Bps

T (mseg)

Primeiro harmônico (Hz)

Harmônicos enviados

300

26,67

37,5

80

600

13,33

75

40

1200

6,67

150

20

2.400

3,33

300

10

4800

1,67

600

5

9600

0,83

1200

2

19200

0,42

2.400

1

38400

0,21

4800

0

Figura 2-2. Relação entre taxa de dados e harmônicos para nosso exemplo.

Há muita confusão sobre largura de banda porque significa coisas diferentes para engenheiros elétricos e cientistas da computação. Para engenheiros elétricos, (análogo) a largura de banda é (como descrevemos acima) uma quantidade medida em Hz. Para cientistas de computadores, a largura de banda (digital) é a taxa máxima de dados de um canal, quantidade medida em bits / s. Essa taxa de dados é o resultado final do uso do analógico largura de banda de um canal físico para transmissão digital, e os dois estão relacionados, como discutiremos a seguir. Neste livro, ficará claro a partir do contexto se nós largura de banda analógica média (Hz) ou largura de banda digital (bits / s).

2.1.3 A taxa máxima de dados de um canal

Já em 1924, um engenheiro da AT&T, Henry Nyquist, percebeu que mesmo um canal certo tem uma capacidade de transmissão finita. Ele derivou uma equação que expressa a taxa de dados máxima para um canal sem ruído de largura de banda finita. Em 1948, Claude Shannon levou o trabalho de Nyquist adiante e o estendeu ao caso de um canal sujeito a ruído aleatório (isto é, termodinâmico) (Shannon, 1948). Este papel é o papel mais importante em toda a teoria da informação. Vamos apenas resumir brevemente marize seus resultados agora clássicos aqui.

Nyquist provou que, se um sinal arbitrário foi executado através de um filtro passa-baixo da largura de banda B , o sinal filtrado pode ser completamente reconstruído, tornando apenas 2 amostras B (exatas) por segundo. Amostragem da linha mais rápido do que $2B$ vezes por o segundo é inútil porque os componentes de maior frequência que tal amostragem poderia recuperar já foram filtrados. Se o sinal consiste em V discreto níveis, o teorema de Nyquist afirma:

$$\text{taxa máxima de dados} = 2B \log_2 V \text{ bits / s}$$

(2-2)

Por exemplo, um canal silencioso de 3 kHz não pode transmitir binários (ou seja, dois níveis) sinais a uma taxa superior a 6.000 bps.

Até agora, consideramos apenas canais sem ruído. Se houver ruído aleatório ent, a situação se deteriora rapidamente. E sempre há ruído aleatório (térmico) presente devido ao movimento das moléculas no sistema. A quantidade de energia térmica o ruído presente é medido pela relação entre a potência do sinal e a potência do ruído. ed o **SNR** (**relação sinal-ruído**). Se denotarmos a potência do sinal por S e o potência de ruído por N , a relação sinal-para-ruído é S/N . Normalmente, a proporção é expressa em uma escala logarítmica como a quantidade $10 \log_{10} S/N$ porque pode variar ao longo de um tremendo alcance. As unidades desta escala logarítmica são chamadas de **decibéis (dB)**, com " deci " significando

10 e " bel " escolhidos para homenagear Alexander Graham Bell, que inventou o telefone. Uma proporção S/N de 10 é 10 dB, uma proporção de 100 é 20 dB, uma proporção de 1000 é 30

dB e assim por diante. Os fabricantes de amplificadores estéreo costumam caracterizar o largura de banda (faixa de frequência) sobre a qual seus produtos são lineares, dando o 3 freqüência dB em cada extremidade. Estes são os pontos em que o fator de amplificação foi reduzido aproximadamente à metade (porque $10 \log_{10} 0,5 \sim -3$).

O principal resultado de Shannon é que a taxa máxima de dados ou **capacidade** de um barulhento canal cuja largura de banda é B Hz e cuja relação sinal-ruído é S/N , é dado por:

$$\text{número máximo de bits / s} = B \log_2 (1 + S/N)$$

(2-3)

Isso nos mostra as melhores capacidades que os canais reais podem ter. Por exemplo, ADSL (Asymmetric Digital Subscriber Line), que fornece acesso à Internet sobre linhas telefônicas ruins, usa uma largura de banda de cerca de 1 MHz. O SNR depende fortemente na distância da casa da central telefônica, e um SNR de cerca de 40 dB para linhas curtas de 1 a 2 km é muito bom. Com essas características,

o canal nunca pode transmitir muito mais do que 13 Mbps, não importa quantos ou quão poucos níveis de sinal são usados e não importa quantas vezes ou quão raramente samples são tomados. Na prática, o ADSL é especificado em até 12 Mbps, embora os usuários frequentemente veja taxas mais baixas. Esta taxa de dados é realmente muito boa, com mais de 60 anos de

técnicas de comunicação que reduziram muito a lacuna entre os canais de Shannon capacidade e capacidade dos sistemas reais.

O resultado de Shannon foi derivado de argumentos da teoria da informação e aplica a qualquer canal sujeito a ruído térmico. Contra-exemplos devem ser tratados na mesma categoria das máquinas de movimento perpétuo. Para ADSL exceder 13 Mbps, deve melhorar o SNR (por exemplo, inserindo repetidores digitais no linhas mais próximas dos clientes) ou usar mais largura de banda, como é feito com o evolução para ASDL2 +.

2.2 MEIOS DE TRANSMISSÃO GUIADA

O objetivo da camada física é transportar bits de uma máquina para uma de outros. Vários meios físicos podem ser usados para a transmissão real. Cada um tem seu próprio nicho em termos de largura de banda, atraso, custo e facilidade de instalação e manutenção. Os meios são agrupados aproximadamente em meios guiados, como fio de cobre e fibra ótica e mídia não guiada, como sem fio terrestre, satélite e lasers no ar. Veremos a mídia guiada nesta seção e não guiada mídia nas próximas seções.

2.2.1 Mídia Magnética

Uma das maneiras mais comuns de transportar dados de um computador para outro é gravá-los em fita magnética ou mídia removível (por exemplo, DVDs graváveis), transportar fisicamente a fita ou os discos para a máquina de destino e lê-los de volta. Embora este método não seja tão sofisticado quanto usar um geosatélite de comunicação crônico, muitas vezes é mais econômico, especialmente para aplicações nas quais a alta largura de banda ou custo por bit transportado é o fator chave. Um cálculo simples tornará este ponto claro. Um Ultrium padrão da indústria fita pode conter 800 gigabytes. Uma caixa de $60 \times 60 \times 60$ cm pode conter cerca de 1000 destes fitas, para uma capacidade total de 800 terabytes ou 6400 terabits (6,4 petabits). Uma caixa de fitas podem ser entregues em qualquer lugar nos Estados Unidos em 24 horas pelo Federal Express e outras empresas. A largura de banda efetiva desta transmissão é 6400 terabits / 86.400 seg, ou um pouco mais de 70 Gbps. Se o destino for apenas uma hora longe por estrada, a largura de banda é aumentada para mais de 1700 Gbps. Sem rede de computador o trabalho pode até se aproximar disso. Claro, as redes estão ficando mais rápidas, mas a fita as cidades também estão aumentando. Se olharmos agora para o custo, teremos uma imagem semelhante. O custo de uma fita Ultrium custa cerca de US \$ 40 quando comprado a granel. Uma fita pode ser reutilizada pelo menos 10 vezes, então o

Página 120

96

A CAMADA FÍSICA
INDIVÍDUO. 2

o custo da fita é talvez \$ 4000 por caixa por uso. Adicione a isso outros \$ 1000 para envio-ping (provavelmente muito menos), e temos um custo de aproximadamente US \$ 5.000 para enviar 800 TB.

Isso equivale a enviar um gigabyte por pouco mais de meio centavo. Nenhuma rede pode vencer isso. A moral da história é:

Nunca subestime a largura de banda de uma perua cheia de fitas correndo pela rodovia.

2.2.2 Pares Trançados

Embora as características da largura de banda da fita magnética sejam excelentes, as características leigas são pobres. O tempo de transmissão é medido em minutos ou horas, não milissegundos. Para muitas aplicações, é necessária uma conexão online. Um de o meio de transmissão mais antigo e ainda mais comum é o **par trançado**. Um torcido O par consiste em dois fios de cobre isolados, normalmente com cerca de 1 mm de espessura. Os fios são torcidos juntos em uma forma helicoidal, assim como uma molécula de DNA. Torcer está feito porque dois fios paralelos constituem uma antena fina. Quando os fios estão torcidos,

as ondas de diferentes torções se cancelam, de modo que o fio irradia com menos eficácia. UMA sinal é normalmente transportado como a diferença de voltagem entre os dois fios no par. Isso fornece melhor imunidade ao ruído externo porque o ruído tende a afetar ambos os fios da mesma forma, deixando o diferencial inalterado.

A aplicação mais comum do par trançado é o sistema telefônico.

Quase todos os telefones são conectados ao escritório da companhia telefônica (telco) por um par trançado. Tanto as chamadas telefônicas quanto o acesso ADSL à Internet passam por essas linhas.

Os pares trançados podem percorrer vários quilômetros sem amplificação, mas por longos períodos pressões que o sinal se torna muito atenuado e são necessários repetidores. Quando muitos pares trançados correm em paralelo por uma distância substancial, como todos os fios que chegam de um prédio de apartamentos para o escritório da companhia telefônica, eles são agrupados para-juntos e envolto em uma bainha protetora. Os pares nestes pacotes interagem fere um com o outro se não fosse pela torção. Em partes do mundo onde linhas telefônicas correm em postes acima do solo, é comum ver vários pacotes centímetros de diâmetro.

Os pares trançados podem ser usados para transmitir informações analógicas ou digitais.

A largura de banda depende da espessura do fio e da distância percorrida, mas vários megabits / s podem ser alcançados por alguns quilômetros em muitos casos. Devido a seu desempenho adequado e baixo custo, pares trançados são amplamente utilizados e são provavelmente permanecerá assim por muitos anos.

O cabeamento de par trançado vem em diversas variedades. A variedade do jardim implantada em muitos edifícios de escritórios é chamado **de cabeamento de categoria 5** ou "Cat 5." A categoria 5

O par trançado consiste em dois fios isolados suavemente torcidos juntos. Quatro dessas pares são normalmente agrupados em uma bainha de plástico para proteger os fios e mantê-los juntos. Esse arranjo é mostrado na Fig. 2-3.

Diferentes padrões de LAN podem usar os pares trançados de maneira diferente. Por exemplo, A Ethernet de 100 Mbps usa dois (dos quatro) pares, um par para cada direção.

Página 121

SEC. 2,2

MÍDIA DE TRANSMISSÃO GUIADA

97

Par trançado

Figura 2-3. Cabo UTP de categoria 5 com quatro pares trançados.

Para alcançar velocidades mais altas, a Ethernet de 1 Gbps usa todos os quatro pares em ambas as direções simultaneamente; isso requer que o receptor fatorar o sinal que é transmitido localmente.

Alguma terminologia geral está em ordem. Links que podem ser usados tanto em duas seções ao mesmo tempo, como uma estrada de duas pistas, são chamadas **de links full-duplex**. No entanto, links que podem ser usados em qualquer direção, mas apenas de uma forma por vez, como

uma linha férrea de via única, são chamados **de links half-duplex**. Uma terceira categoria consiste em links que permitem o tráfego em apenas uma direção, como uma rua de mão única. Eles são chamados de links **simplex**.

Voltando ao par trançado, o Cat 5 substituiu os cabos **Categoria 3** anteriores por um cabo semelhante que usa o mesmo conector, mas tem mais torções por metro. Mais torções resultam em menos diafonia e um sinal de melhor qualidade em distâncias mais longas, tornando os cabos mais adequados para comunicação de computador de alta velocidade, especialmente

principalmente LANs Ethernet de 100 Mbps e 1 Gbps.

É mais provável que a nova fiação seja da **Categoria 6** ou mesmo da **Categoria 7**. Estes categorias tem especificações mais rigorosas para lidar com sinais com banda maior larguras. Alguns cabos na Categoria 6 e acima são classificados para sinais de 500 MHz

e pode suportar os links de 10 Gbps que serão implantados em breve. Por meio da Categoria 6, esses tipos de fiação são referidos como **UTP** (sem **blindagem Par trançado**), pois consistem simplesmente de fios e isoladores. Em contraste com estes, Os cabos da categoria 7 têm blindagem nos pares trançados individuais, bem como ao redor todo o cabo (mas dentro da capa protetora de plástico). A blindagem reduz o suscetibilidade a interferência externa e diafonia com outros cabos próximos para atender às exigentes especificações de desempenho. Os cabos são uma reminiscência dos cabos de par trançado blindado de alta qualidade, mas volumosos e caros que a IBM introduziu produzido no início dos anos 1980, mas que não se provou popular fora da IBM, instalações. Evidentemente, é hora de tentar novamente.

2.2.3 Cabo Coaxial

Outro meio de transmissão comum é o **cabo coaxial** (conhecido por seu muitos amigos apenas "persuadem" e pronunciam-se "coaxiais"). Tem melhor blindagem e maior largura de banda do que os pares trançados não blindados, para que possa abranger distâncias mais longas em

Página 122

98

A CAMADA FÍSICA
INDIVÍDUO. 2

velocidades mais altas. Dois tipos de cabo coaxial são amplamente usados. Um tipo, 50 ohm cabo, é comumente usado quando se destina à transmissão digital do começar. O outro tipo, o cabo de 75 ohms, é comumente usado para transmissão analógica e televisão a cabo. Esta distinção é baseada no histórico, ao invés de técnico, fatores (por exemplo, as primeiras antenas dipolo tinham uma impedância de 300 ohms, e era fáceis de usar os transformadores de casamento de impedância 4: 1 existentes). Começando no meio Década de 1990, as operadoras de TV a cabo começaram a fornecer acesso à Internet por cabo, que tem

tornou o cabo de 75 ohms mais importante para a comunicação de dados.

Um cabo coaxial consiste em um fio de cobre rígido como núcleo, rodeado por um material isolante. O isolador é envolvido por um condutor cilíndrico, muitas vezes como um malha trançada intimamente tecida. O condutor externo é coberto por um plástico protetor bainha de tique. Uma vista em corte de um cabo coaxial é mostrada na Figura 2-4.

Cobre
testemunho
Isolante
material
Trançado
exterior
condutor
Protetora
plástico
cobertura

Figura 2-4. Um cabo coaxial.

A construção e blindagem do cabo coaxial proporcionam uma boa combinação de alta largura de banda e excelente imunidade a ruído. A largura de banda possível depende da qualidade e do comprimento do cabo. Os cabos modernos têm uma largura de banda de até um

poucos GHz. Os cabos coaxiais costumavam ser amplamente utilizados no sistema telefônico para linhas de longa distância, mas agora foram amplamente substituídas por fibra óptica em longas rotas de transporte. Coax ainda é amplamente utilizado para televisão a cabo e áreas metropolitanas redes, no entanto.

2.2.4 Linhas de energia

As redes de telefonia e televisão a cabo não são as únicas fontes de wir- que pode ser reutilizado para comunicação de dados. Existe um tipo ainda mais comum de fiação: linhas de energia elétrica. As linhas de força fornecem energia elétrica para as casas, e a fiação elétrica dentro das casas distribui a energia para as tomadas elétricas.

O uso de linhas de energia para comunicação de dados é uma ideia antiga. Linhas de energia têm sido usados por empresas de eletricidade para comunicação de baixa taxa, como remediação de mtoe por muitos anos, bem como em casa para controlar dispositivos (por exemplo, o

Padrão X10). Nos últimos anos, tem havido um interesse renovado em altas taxas de juros. comunicação através dessas linhas, tanto dentro de casa como uma LAN e fora de casa

Página 123

SEC. 2,2
MÍDIA DE TRANSMISSÃO GUIADA

99

para acesso à Internet de banda larga. Vamos nos concentrar no cenário mais comum: usando fios elétricos dentro de casa.

A conveniência de usar linhas de energia para rede deve ser clara. Simplesmente conecte uma TV e um receptor na parede, o que você deve fazer de qualquer maneira, porque eles precisam de energia e podem enviar e receber filmes pela fiação elétrica. Isto a configuração é mostrada na Figura 2-5. Não há outro plugue ou rádio. O sinal de dados é sobreposto ao sinal de energia de baixa frequência (no ativo ou "quente" fio), pois ambos os sinais usam a fiação ao mesmo tempo.

Sinal de energia
Sinal de dados
Cabo elétrico

Figura 2-5. Uma rede que usa fiação elétrica doméstica.

A dificuldade em usar fiação elétrica doméstica para uma rede é que foi projetado para distribuir sinais de energia. Esta tarefa é bastante diferente da distribuição mascar sinais de dados, nos quais a fiação doméstica faz um péssimo trabalho. Sinal elétrico nais são enviados em 50-60 Hz e a fiação atenua a frequência muito mais alta (MHz) sinais necessários para comunicação de dados de alta taxa. As propriedades elétricas da fiação variam de uma casa para a próxima e mudam conforme os aparelhos são ligados ligado e desligado, o que faz com que os sinais de dados saltem ao redor da fiação. Transient curtos aluguéis quando os aparelhos são ligados e desligados criam ruído elétrico em uma ampla faixa de frequências. E sem a torção cuidadosa de pares trançados, fiação elétrica atua como uma antena fina, captando sinais externos e irradiando sinais próprios.

Esse comportamento significa que para atender aos requisitos regulamentares, o sinal de dados deve excluir as frequências licenciadas, como as bandas de rádio amador.

Apesar dessas dificuldades, é prático enviar pelo menos 100 Mbps além do normal fiação elétrica doméstica usando esquemas de comunicação que resistem aos deficientes freqüências e rajadas de erros. Muitos produtos usam vários padrões proprietários para redes de linha de energia, portanto, os padrões internacionais estão ativamente em desenvolvimento

ment.

2.2.5 Fibra Óptica

Muitas pessoas na indústria de computadores se orgulham da rapidez com que a tecnologia de computador está melhorando à medida que segue a lei de Moore, que prevê um duplo bling do número de transistores por chip aproximadamente a cada dois anos (Schaller,

Página 124

100
A CAMADA FÍSICA
INDIVÍDUO. 2

1997). O IBM PC original (1981) funcionava a uma velocidade de clock de 4,77 MHz. Vinte-oito anos depois, os PCs podiam rodar uma CPU de quatro núcleos a 3 GHz. Este aumento é um ganho de um fator de cerca de 2500, ou 16 por década. Impressionante.

No mesmo período, os links de comunicação de área ampla passaram de 45 Mbps (um T3 linha no sistema telefônico) a 100 Gbps (uma moderna linha de longa distância). Isto o ganho é igualmente impressionante, mais do que um fator de 2.000 e perto de 16 por década, enquanto ao mesmo tempo a taxa de erro passou de 10

-5

por pouco a quase zero. Além disso, CPUs únicas estão começando a se aproximar dos limites físicos, que

é por isso que agora é o número de CPUs que está sendo aumentado por chip. Em contraste, a largura de banda alcançável com tecnologia de fibra é superior a 50.000 Gbps (50 Tbps) e não estamos nem perto de atingir esses limites. O limite prático atual de cerca de 100 Gbps é devido à nossa incapacidade de converter entre elétrico e óptico sinais de cal mais rápido. Para construir links de maior capacidade, muitos canais são simplesmente transportados em paralelo em uma única fibra.

Nesta seção, estudaremos a fibra óptica para aprender como essa tecnologia de transmissão funciona. Na corrida contínua entre computação e comunicação, com comunicação ainda pode vencer por causa das redes de fibra óptica. A implicação disso seria essencialmente largura de banda infinita e uma nova sabedoria convencional que computadores são irremediavelmente lentos, de modo que as redes devem tentar evitar a computação custos, não importa quanta largura de banda seja desperdiçada. Essa mudança vai demorar um pouco para mergulhar em uma geração de cientistas da computação e engenheiros ensinados a pensar em termos dos limites baixos de Shannon impostos pelo cobre.

Claro, este cenário não conta toda a história porque não clude custo. O custo de instalação de fibra na última milha para alcançar os consumidores e ignorar a baixa largura de banda dos fios e a disponibilidade limitada de espectro é tremenda dous. Também custa mais energia mover bits do que calcular. Podemos sempre têm ilhas de desigualdades onde computação ou comunicação é essencial oficialmente grátis. Por exemplo, no limite da Internet, lançamos computação e armazenamento no problema de compactação e armazenamento em cache de conteúdo, tudo para fazer um melhor uso de links de acesso à Internet. Na Internet, podemos fazer o inverso, com empresas como o Google movendo enormes quantidades de dados pela rede para onde é mais barato armazenar ou computar nele.

A fibra óptica é usada para transmissão de longa distância em backbones de rede, LANs de velocidade (embora, até agora, o cobre sempre tenha conseguido alcançar eventualmente), e acesso de alta velocidade à Internet, como **FttH** (**Fiber to the Home**). Um óptico sistema de transmissão tem três componentes principais: a fonte de luz, a transmissão meio e o detector. Convencionalmente, um pulso de luz indica 1 bit e a ausência de luz indica um bit 0. O meio de transmissão é ultrafino fibra de vidro. O detector gera um pulso elétrico quando a luz incide sobre ele. Por anexar uma fonte de luz a uma extremidade de uma fibra óptica e um detector à outra, temos um sistema de transmissão de dados unidirecional que aceita um sinal elétrico final, converte e transmite por pulsos de luz e, em seguida, reconverte a saída para um sinal elétrico na extremidade receptora.

Página 125

SEC. 2,2 MÍDIA DE TRANSMISSÃO GUIADA

101

Este sistema de transmissão vazaria luz e seria inútil na prática se fosse não por um interessante princípio de física. Quando um raio de luz passa de um meio para outro - por exemplo, de sílica fundida ao ar - o raio é refratado (dobrado) na fronteira sílica / ar, conforme mostrado na Fig. 2-6 (a). Aqui vemos um raio de luz incidente na fronteira com um ângulo α_1 emergindo com um ângulo β_1 . A quantidade de refração depende das propriedades dos dois meios (em particular, seus índices de refração). Para ângulos de incidência acima de um certo valor crítico, a luz é refratada de volta na sílica; nada disso escapa no ar. Assim, um raio de luz incidente no ângulo crítico ou acima dele fica preso dentro da fibra, conforme mostrado em Figura 2-6 (b) e pode se propagar por muitos quilômetros praticamente sem perdas.

Interno total reflexão.
Ar / sílica fronteira
Fonte de luz
Sílica
Ar
(uma)
(b)

β_1

β_2

β_3

α_1

α_2

α_3

Figura 2-6. (a) Três exemplos de um raio de luz de dentro de uma fibra de sílica colidindo na fronteira ar / sílica em diferentes ângulos. (b) Luz capturada por inter-reflexão final.

O esboço da Fig. 2-6 (b) mostra apenas um raio preso, mas como qualquer raio de luz incidente no limite acima do ângulo crítico será refletido internamente, muitos raios diferentes estarão refletindo em diferentes ângulos. Cada raio é dito ter um modo diferente, então uma fibra com essa propriedade é chamada de **multimodo fibra**.

No entanto, se o diâmetro da fibra for reduzido a alguns comprimentos de onda de luz, o a fibra atua como um guia de onda e a luz pode se propagar apenas em linha reta, sem saltar, produzindo uma **fibra monomodo**. As fibras monomodo são mais expensivas, mas são amplamente utilizadas para distâncias mais longas. Modo único disponível atualmente

as fibras podem transmitir dados a 100 Gbps por 100 km sem amplificação. Até taxas de dados mais altas foram alcançadas no laboratório para distâncias mais curtas.

Transmissão de luz através da fibra

As fibras ópticas são feitas de vidro, que, por sua vez, é feito de areia, uma inexistência matéria-prima pensativa disponível em quantidades ilimitadas. A fabricação de vidro era conhecida por os antigos egípcios, mas seu vidro não tinha mais do que 1 mm de espessura ou

102

A CAMADA FÍSICA

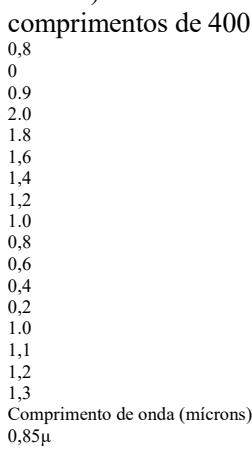
INDIVÍDUO. 2

a luz não podia brilhar. Vidro transparente o suficiente para ser útil para janelas foi desenvolvido durante o Renascimento. O vidro usado para fibras ópticas modernas é tão transparente que se os oceanos estivessem cheios dela em vez de água, o fundo do mar seria tão visível da superfície quanto o solo é de um avião em um claro dia.

A atenuação da luz através do vidro depende do comprimento de onda da luz (bem como em algumas propriedades físicas do vidro). É definido como a proporção de entrada para potência do sinal de saída. Para o tipo de vidro usado nas fibras, a atenuação é mostrado na Figura 2-7 em unidades de decibéis por quilômetro linear de fibra. Para a prova- ple, um fator de dois a perda de potência do sinal dá uma atenuação de $10 \log_{10} 2 = 3 \text{ dB}$.

A figura mostra a parte do infravermelho próximo do espectro, que é o que é usado em prática. A luz visível tem comprimentos de onda ligeiramente mais curtos, de 0,4 a 0,7 micrôn. (1 micrôn é 10^{-6} metros.) O verdadeiro purista métrico se referiria a essas ondas

comprimentos de 400 nm a 700 nm, mas continuaremos com o uso tradicional.



Banda
1,30 μ
Banda
1,55 μ
Banda
Atenuação (dB / km)
1,4
1,5
1,6
1,7
1,8

Figura 2-7. Atenuação da luz através da fibra na região do infravermelho.

Três bandas de comprimento de onda são mais comumente usadas no momento para comunicação. Eles são centralizados em 0,85, 1,30 e 1,55 mícrons, respectivamente. Todos três bandas têm 25.000 a 30.000 GHz de largura. A banda de 0,85 micron foi usada primeiro. Tem maior atenuação e, portanto, é usado para distâncias mais curtas, mas nessa onda comprimento os lasers e eletrônicos podem ser feitos do mesmo material (gálio arseneto). As duas últimas bandas têm boas propriedades de atenuação (menos de 5% de perda por quilômetro). A banda de 1,55 micron é agora amplamente usada com dopagem de érbio amplificadores que atuam diretamente no domínio óptico.

Página 127

SEC. 2,2

MÍDIA DE TRANSMISSÃO GUIADA

103

Pulsos de luz enviados por uma fibra espalham-se em comprimento à medida que se propagam. Isto espalhar é chamado de **dispersão cromática**. A quantidade disso depende do comprimento de onda dente. Uma maneira de evitar que esses pulsos espalhados se sobreponham é aumentar o distância entre eles, mas isso só pode ser feito reduzindo a taxa de sinalização.

Felizmente, foi descoberto que fazer os pulsos em uma forma especial se relacionam com o recíproco do cosseno hiperbólico causa quase todo o efeito de dispersão efeitos se cancelam, então é possível enviar pulsos por milhares de quilômetros com distorção de forma apreciável. Esses pulsos são chamados de **solitons**. Um considerável muitas pesquisas estão em andamento para tirar os solitons do laboratório e colocá-los no campo.

Cabos de fibra

Os cabos de fibra óptica são semelhantes aos coaxiais, exceto sem a trança. Figura 2-8 (a) mostra uma única fibra vista de lado. No centro está o núcleo de vidro através qual a luz se propaga. Em fibras multimodo, o núcleo é normalmente de 50 mícrons de diâmetro, aproximadamente da espessura de um cabelo humano. Em fibras monomodo, o núcleo é de 8 a 10 mícrons.

Jaqueta
(plástico)
Testemunho
Cladding
Bainha
Jaqueta
Cladding
(vidro)
Testemunho
(vidro)
(uma)
(b)

Figura 2-8. (a) Vista lateral de uma única fibra. (b) Vista final de uma bainha com três fibras.

O núcleo é cercado por um revestimento de vidro com um índice de refração inferior do que o núcleo, para manter toda a luz no núcleo. Em seguida, vem uma fina jaqueta de plástico para

proteger o revestimento. As fibras são normalmente agrupadas em pacotes, protegidas por um bainha externa. A Figura 2-8 (b) mostra uma bainha com três fibras.

Bainhas de fibra terrestre são normalmente colocadas no solo dentro de um metro do superfície, onde ocasionalmente estão sujeitos a ataques de retroescavadeiras ou esquilos.

Perto da costa, bainhas de fibras transoceânicas são enterradas em trincheiras por uma espécie de seaplow. Em águas profundas, eles apenas ficam no fundo, onde podem ser agarrados por traineiras de pesca ou atacadas por lulas gigantes.

As fibras podem ser conectadas de três maneiras diferentes. Primeiro, eles podem terminar em

conectores e ser conectado a soquetes de fibra. Os conectores perdem cerca de 10 a 20% de a luz, mas tornam mais fácil reconfigurar os sistemas. Em segundo lugar, eles podem ser emendados mecanicamente. Emendas mecânicas apenas colocam o duas pontas cuidadosamente cortadas lado a lado em uma manga especial e prendê-las

Página 128

104

A CAMADA FÍSICA INDIVÍDUO. 2

Lugar, colocar. O alinhamento pode ser melhorado passando luz através da junção e então fazendo pequenos ajustes para maximizar o sinal. Emendas mecânicas tomam trem editado cerca de 5 minutos e resultar em 10% de perda de luz.

Terceiro, dois pedaços de fibra podem ser fundidos (derretidos) para formar uma conexão sólida. UMA

a emenda de fusão é quase tão boa quanto uma única fibra desenhada, mas mesmo aqui, um pequeno quantidade de atenuação ocorre.

Para todos os três tipos de emendas, podem ocorrer reflexos no ponto da emenda, e a energia refletida pode interferir no sinal.

Dois tipos de fontes de luz são normalmente usados para fazer a sinalização. Esses são LEDs (diodos emissores de luz) e lasers semicondutores. Eles têm diferentes propriedades, conforme mostrado na Fig. 2-9. Eles podem ser ajustados no comprimento de onda inserindo

Interferômetros Fabry-Perot ou Mach-Zehnder entre a fonte e a fibra.

Os interferômetros Fabry-Perot são cavidades ressonantes simples que consistem em dois espelhos. A luz incide perpendicularmente aos espelhos. O comprimento da caverna ity seleciona os comprimentos de onda que cabem dentro de um número inteiro de vezes.

Os interferômetros Mach-Zehnder separam a luz em dois feixes. Os dois feixes viajar distâncias ligeiramente diferentes. Eles são recombinados no final e estão em fase para apenas certos comprimentos de onda.

Item

CONDUZIU

Laser semicondutor

Taxa de dados

Baixo

Alto

Tipo de fibra

Multi-modo

Multi-modo ou modo único

Distância

Curto

Longo

Tempo de vida

Vida longa

Vida curta

Sensibilidade à temperatura

Menor

Substancial

Custo

Baixo custo

Caro

Figura 2-9. Uma comparação de diodos semicondutores e LEDs como fontes de luz.

A extremidade receptora de uma fibra óptica consiste em um fotodíodo, que emite um pulso elétrico quando atingido pela luz. O tempo de resposta dos fotodiodos, que converter o sinal do domínio ótico para o elétrico, limitar as taxas de dados para cerca de 100 Gbps. O ruído térmico também é um problema, portanto, um pulso de luz deve levar energia suficiente para ser detectada. Ao tornar os pulsos fortes o suficiente, o erro a taxa pode ser arbitrariamente pequena.

Comparação de fibra óptica e fio de cobre

É instrutivo comparar a fibra com o cobre. A fibra tem muitas vantagens. Para para começar, ele pode lidar com larguras de banda muito maiores do que o cobre. Só isso seria

exigem seu uso em redes de ponta. Devido à baixa atenuação, os repetidores são necessários apenas a cada 50 km em linhas longas, contra cerca de 5 km no cobre,

Página 129

SEC. 2,2

MÍDIA DE TRANSMISSÃO GUIADA

105

resultando em uma grande economia de custos. A fibra também tem a vantagem de não ser afetada por picos de energia, interferência eletromagnética ou falhas de energia. Nem é afetada por produtos químicos corrosivos no ar, importantes para ambientes industriais agressivos. Curiosamente, as empresas de telefonia gostam de fibra por um motivo diferente: é fina e leve. Muitos dutos de cabos existentes estão completamente cheios, então não há espaço para adicionar nova capacidade. Removendo todo o cobre e substituindo-o por fibra esvazia os dutos, e o cobre tem excelente valor de revenda para refinadores de cobre que o vêem como minério de muito alto teor. Além disso, a fibra é muito mais leve que o cobre. 1 mil pares trançados de 1 km de comprimento pesam 8.000 kg. Duas fibras têm mais capacidade e pesam apenas 100 kg, o que reduz a necessidade de caro suporte mecânico sistemas que devem ser mantidos. Para novas rotas, a fibra ganha sem dúvida devida a seu custo de instalação muito mais baixo. Finalmente, as fibras não vazam luz e são difíceis de tocar. Essas propriedades fornecem à fibra uma boa segurança contra possíveis grampos telefônicos. Por outro lado, a fibra é uma tecnologia menos familiar que requer habilidades nem todos os engenheiros sim, e as fibras podem ser danificadas facilmente por serem dobradas demais. Desde o ponto de vista da transmissão óptica é inherentemente unidirecional, a comunicação bidirecional requer duas fibras ou duas bandas de frequência em uma fibra. Finalmente, interfaces de fibra costumam mais do que interfaces elétricas. No entanto, o futuro de todos os dados fixos é a comunicação em distâncias mais curtas é claramente feita com fibra. Para um discussão de todos os aspectos da fibra óptica e suas redes, ver Hecht (2005).

2.3 TRANSMISSÃO SEM FIO

Nossa era deu origem a viciados em informação: pessoas que precisam estar online o tempo todo. Para esses usuários móveis, par trançado, coaxial e fibra óptica não são opções. Eles precisam obter seus "resultados" de dados para seu laptop, notebook, bolso de camisa, palmtop ou relógios de pulso sem estar conectado ao computador terrestre. Infraestrutura de comunicação. Para esses usuários, a comunicação sem fio é a resposta.

Nas seções a seguir, veremos a comunicação sem fio em geral.

Ele tem muitos outros aplicativos importantes, além de fornecer conectividade aos usuários que querem navegar na Web da praia. Wireless tem vantagens até mesmo para fixos dispositivos em algumas circunstâncias. Por exemplo, se passar uma fibra para um edifício é difícil devido ao terreno (montanhas, selvas, pântanos, etc.), o wireless pode ser uma alternativa. Vale ressaltar que a comunicação digital sem fio moderna começou nas Ilhas Havaianas, onde grandes pedaços do Oceano Pacífico separavam os usuários de seu centro de informática e o sistema telefônico eram inadequados.

2.3.1 O Espectro Eletromagnético

Quando os elétrons se movem, eles criam ondas eletromagnéticas que podem se propagar através do espaço (mesmo no vácuo). Essas ondas foram previstas pelos britânicos físicos James Clerk Maxwell em 1865 e observado pela primeira vez pelo alemão

Página 130

106

A CAMADA FÍSICA

INDIVÍDUO. 2

físico Heinrich Hertz em 1887. O número de oscilações por segundo de uma onda é chamada de **frequência**, f , e é medida em **Hz** (em homenagem a Heinrich Hertz). A distância entre dois máximos (ou mínimos) consecutivos é chamada de **comprimento de onda**, que é universalmente designado pela letra grega λ (lambda).

Quando uma antena de tamanho apropriado é conectada a um circuito elétrico, as ondas eletromagnéticas podem ser transmitidas de forma eficiente e recebidas por um receptor

alguma distância. Toda a comunicação sem fio é baseada neste princípio.

No vácuo, todas as ondas eletromagnéticas viajam na mesma velocidade, não importa qual a sua frequência. Essa velocidade, geralmente chamada de **velocidade da luz**, c , é de aproximadamente 3×10^8 m / s, ou cerca de 1 pé (30 cm) por nanosegundo. (Um caso pode ser feito para redefinir o pé conforme a distância que a luz viaja no vácuo em 1 nseg em vez de basear-se no tamanho do sapato de algum rei morto há muito tempo.) Em cobre ou fibra a velocidade diminui para cerca de 2/3 deste valor e torna-se ligeiramente dependente da frequência dente. A velocidade da luz é o limite de velocidade final. Nenhum objeto ou sinal pode mover mais rápido do que ele.

A relação fundamental entre f , λ e c (no vácuo) é

$$\lambda f = c$$

(2-4)

Como c é uma constante, se conhecermos f , podemos encontrar λ e vice-versa. Como regra de polegar, quando λ está em metros e f está em MHz, $\lambda f \sim 300$. Por exemplo, 100 MHz as ondas têm cerca de 3 metros de comprimento, ondas de 1000 MHz têm 0,3 metros de comprimento e 0,1-

ondas métricas têm uma frequência de 3000 MHz.

O espectro eletromagnético é mostrado na Figura 2-10. O rádio, microondas, infravermelho e porções de luz visível do espectro podem ser usados para transmitir informações modulando a amplitude, frequência ou fase das ondas.

A luz ultravioleta, os raios X e os raios gama seriam ainda melhores, devido à sua alta mais frequências, mas são difíceis de produzir e modular, não se propagam bem através de edifícios e são perigosos para os seres vivos. As bandas listadas no botom da Fig. 2-10 são os oficiais ITU (International Telecommunication Union) nomes e são baseados nos comprimentos de onda, então a banda LF vai de 1 km a 10 km (aproximadamente 30 kHz a 300 kHz). Os termos LF, MF e HF referem-se a Baixo, Média e alta frequência, respectivamente. Claramente, quando os nomes eram tão assinado, ninguém esperava ir acima de 10 MHz, então as bandas mais altas foram mais tarde chamada de frequência muito, ultra, super, extremamente e tremendamente alta bandas. Além disso, não há nomes, mas Incredibly, Astonishingly, e Prod-Frequentemente alta (IHF, AHF e PHF) soaria bem.

Sabemos de Shannon [Eq. (2-3)] que a quantidade de informações que um sinal final, como uma onda eletromagnética pode transportar depende da potência recebida e é proporcional à sua largura de banda. Da Fig. 2-10, agora deve ser óbvio por que pessoas em rede gostam muito de fibra óptica. Muitos GHz de largura de banda estão disponíveis capaz de tocar para transmissão de dados na banda de micro-ondas e ainda mais em fibra porque está mais à direita em nossa escala logarítmica. Por exemplo, considere a banda de 1,30 micron da Fig. 2-7, que tem uma largura de 0,17 micron. Se usarmos

luz
 10₄
 10₅
 10₆
 10₇
 10₈
 10₉
 10₁₀
 10₁₁
 10₁₂
 10₁₃
 10₁₄
 10₁₅
 10₁₆
 f (Hz)
 Par trançado
 Coaxial
 Satélite
 televisão
 Terrestre
 microondas
 Fibra
 ótica
 Marítimo
 SOU
 rádio
 FM
 rádio
 Banda
 LF
 MF
 HF
 VHF
 UHF
 SHF
 EHF
 THF

Figura 2-10. O espectro eletromagnético e seus usos para comunicação.

Eq. (2-4) para encontrar as frequências iniciais e finais dos comprimentos de onda inicial e final, encontramos a faixa de frequência em cerca de 30.000 GHz. Com um sinal razoável relação ao ruído de 10 dB, isso é 300 Tbps.

A maioria das transmissões usa uma banda de frequência relativamente estreita (ou seja, $\Delta f/f \ll 1$). Eles concentram seus sinais nesta banda estreita para usar o espectro de forma eficiente e obter taxas de dados razoáveis transmitindo com potência suficiente. No entanto, em alguns casos, uma banda mais larga é usada, com três variações. No **salto de frequência espalhar o espectro**, o transmissor salta de frequência em frequência centenas de vezes por segundo. É popular para comunicação militar porque torna transmissões difíceis de detectar e quase impossíveis de bloquear. Também oferece boas resistência ao desvanecimento multipercorso e interferência de banda estreita porque o receptor não ficará preso em uma frequência prejudicada por tempo suficiente para desligar a comunicação. Esta robustez o torna útil para partes lotadas do espectro, como como as bandas ISM, iremos descrever em breve. Esta técnica é usada comercialmente, por exemplo, em Bluetooth e versões anteriores de 802.11.

Como uma curiosa nota de rodapé, a técnica foi inventada pelo sexo austríaco deusa Hedy Lamarr, a primeira mulher a aparecer nua em um filme (a 1933, filme tcheco *Extase*). Seu primeiro marido era um fabricante de armamentos que disse a ela como era fácil bloquear os sinais de rádio usados para controlar torpedos. Quando ela descobriu que ele estava vendendo armas para Hitler, ela ficou horrorizada, disfarçou-se de empregada doméstica para escapar dele e fugiu para Hollywood para continuar carreira como atriz de cinema. Em seu tempo livre, ela inventou o salto de frequência para ajudar o esforço de guerra Aliado. Seu esquema usava 88 frequências, o número de teclas

não conseguiram convencer a Marinha dos Estados Unidos de que sua invenção tinha algum uso prático e nunca recebeu royalties. Apenas anos depois que a patente expirou, venha popular. Uma segunda forma de espectro de propagação, espectro de propagação de **sequência direta**, usa um sequência de código para espalhar o sinal de dados por uma banda de frequência mais ampla. É amplamente usado comercialmente como uma maneira espectralmente eficiente de permitir que vários sinais compartilhem a mesma banda de frequência. Esses sinais podem receber códigos diferentes, um método chamado **CDMA (acesso múltiplo por divisão de código)**, ao qual retornaremos posteriormente neste capítulo. Esse método é mostrado em contraste com o salto de frequência na Figura 2-11. Isto forma a base das redes de telefonia móvel 3G e também é usado em GPS (Global Sistema de posicionamento). Mesmo sem códigos diferentes, especificação de propagação de sequência direta trum, como espectro de dispersão de salto de frequência, pode tolerar inter-banda estreita atenuação da diferença e do multipath porque apenas uma fração do sinal desejado é perdida. É usado nesta função em LANs sem fio 802.11b mais antigas. Para um fascinante e de-história de cauda da comunicação de espalhamento espectral, ver Scholtz (1982).

Ultra Wideband
underlay
(Usuário CDMA com
código diferente)
Direto
seqüência
propagação
espectro
Frequência
saltitar
propagação
espectro
Frequência
(Usuário CDMA com
código diferente)

Figura 2-11. Espalhe o espectro e comunicação de banda ultralarga (UWB).

Um terceiro método de comunicação com uma banda mais larga é **UWB (Ultra-Comunicação WideBand)**. UWB envia uma série de pulsos rápidos, variando suas posições para comunicar informações. As transições rápidas levam a um sinal de que está disperso em uma banda de frequência muito ampla. UWB é definido como sinais de que tem uma largura de banda de pelo menos 500 MHz ou pelo menos 20% da frequência central de sua banda de frequência. O UWB também é mostrado na Figura 2-11. Com tanta banda largura, o UWB tem potencial para se comunicar em altas taxas. Porque é espalhado através de uma ampla banda de frequências, pode tolerar uma quantidade substancial de interferência forte de outros sinais de banda estreita. Tão importante quanto, uma vez que UWB tem muito pouca energia em qualquer frequência quando usado para curto alcance transmissão, não causa interferência prejudicial a outras bandas estreitas sinais de rádio. Diz-se que está **subjacente** aos outros sinais. Esta coexistência pacífica levou à sua aplicação em PANs sem fio que funcionam a até 1 Gbps, embora com o sucesso comercial foi misto. Também pode ser usado para geração de imagens por meio de objetos sólidos objetos (solo, paredes e corpos) ou como parte de sistemas de localização precisa.

As ondas de radiofrequência (RF) são fáceis de gerar, podem viajar por longas distâncias, e podem penetrar edifícios facilmente, por isso são amplamente usados para comunicação, dentro e fora de casa. As ondas de rádio também são omnidirecionais, o que significa que eles viajam em todas as direções da fonte, então o transmissor e o receptor não deve ser cuidadosamente alinhado fisicamente.

Às vezes, o rádio omnidirecional é bom, mas às vezes é ruim. No 1970, a General Motors decidiu equipar todos os seus novos Cadillacs com computador-con freios antibloqueio trollados. Quando o motorista pisou no pedal do freio, o computador Pulsou os freios e desligou em vez de travá-los com força. Um belo dia e O patrulheiro rodoviário de Ohio começou a usar seu novo rádio móvel para ligar para a sede, e, de repente, o Cadillac ao lado dele começou a se comportar como um bronco bravo. Quando o policial parou o carro, o motorista alegou que não havia feito nada e que o carro enlouqueceu.

Eventualmente, um padrão começou a surgir: Cadillacs às vezes enlouqueciam, mas apenas nas principais rodovias de Ohio e apenas quando a Patrulha Rodoviária estava assistindo. Por muito, muito tempo, a General Motors não conseguia entender por que Cadillacs funcionou bem em todos os outros estados e também em estradas secundárias em Ohio. Somente

depois de muito pesquisar, eles descobriram que a fiação do Cadillac fazia uma excelente tenna para a frequência usada pelo novo sistema de rádio da Patrulha Rodoviária de Ohio. As propriedades das ondas de rádio dependem da frequência. Em baixas frequências, ondas de rádio passam bem através de obstáculos, mas a energia cai drasticamente com distância da fonte - pelo menos tão rápido quanto $1/r^2$ no ar - pois a energia do sinal é espalhada mais finamente sobre uma superfície maior. Essa atenuação é chamada de **perda de caminho**. Em

altas frequências, as ondas de rádio tendem a viajar em linha reta e ricochetear em obstáculos. A perda de caminho ainda reduz a potência, embora o sinal recebido possa depender fortemente nas reflexões também. As ondas de rádio de alta frequência também são absorvidas pela chuva e outros obstáculos em maior extensão do que os de baixa frequência. Em todas as frequências, ondas de rádio estão sujeitas à interferência de motores e outros equipamentos elétricos.

É interessante comparar a atenuação das ondas de rádio com a dos sinais em mídia guiada. Com fibra, coaxial e par trançado, o sinal cai na mesma proporção fração por unidade de distância, por exemplo 20 dB por 100m para par trançado. Com rádio, o sinal cai na mesma fração que a distância dobrar, por exemplo 6 dB por duplicação no espaço livre. Este comportamento significa que as ondas de rádio podem viajar longas distâncias e a interferência entre os usuários é um problema. Por este motivo, todos os governos regulam rigidamente o uso de transmissores de rádio, com alguns exemplos notáveis de percepções, que são discutidas posteriormente neste capítulo.

Página 134

110

A CAMADA FÍSICA INDIVÍDUO. 2

Nas bandas VLF, LF e MF, as ondas de rádio seguem o solo, conforme ilustrado na Figura 2-12 (a). Essas ondas podem ser detectadas por talvez 1000 km na parte inferior freqüências, menos nas mais altas. A transmissão de rádio AM usa a banda MF, é por isso que as ondas terrestres das estações de rádio AM de Boston não podem ser ouvidas facilmente em Nova York. As ondas de rádio nessas bandas passam facilmente pelos edifícios, é por isso que os rádios portáteis funcionam em ambientes internos. O principal problema em usar estes

bandas para comunicação de dados é sua baixa largura de banda [consulte a Eq. (2-4)].

erehpsonol
superfície da Terra
superfície da Terra
(uma)
(b)
Terra
onda

Figura 2-12. (a) Nas bandas VLF, LF e MF, as ondas de rádio seguem a curvatura da terra. (b) Na banda de HF, elas ricochetiam na ionosfera.

Nas bandas de HF e VHF, as ondas terrestres tendem a ser absorvidas pela terra.

No entanto, as ondas que atingem a ionosfera, uma camada de partículas carregadas agarram-se à terra a uma altura de 100 a 500 km, são refratados por ela e enviados de volta para terra, conforme mostrado na Figura 2-12 (b). Sob certas condições atmosféricas, os sinais pode saltar várias vezes. Operadores de rádio amadores (radioamadores) usam essas bandas para falar

longa distância. Os militares também se comunicam nas bandas de HF e VHF.

2.3.3 Transmissão de Microondas

Acima de 100 MHz, as ondas viajam em linhas quase retas e, portanto, podem ser estreitamente focado. Concentrar toda a energia em um pequeno feixe por meio de um antena parabólica (como a familiar antena parabólica) dá um sinal muito mais alto-relação ao ruído, mas as antenas de transmissão e recepção devem ser precisas alinhados uns com os outros. Além disso, essa direcionalidade permite várias transmitters alinhados em uma linha para se comunicar com vários receptores em uma linha sem interferência, desde que sejam observadas algumas regras de espaçamento mínimo. Antes da fibra óptica, por décadas essas microondas formaram o coração da tele-distância sistema de transmissão de telefone. Na verdade, a MCI, uma das primeiras concorrentes da AT&T depois disso

foi desregulamentado, construiu todo o seu sistema com comunicação de microondas passando entre torres separadas por dezenas de quilômetros. Até o nome da empresa refletia isso (MCI significa Microwave Communications, Inc.). MCI desde então foi para fibra e por meio de uma longa série de fusões e falências de empresas no O embaralhamento das telecomunicações tornou-se parte da Verizon.

Página 135

SEC. 2,3

TRANSMISSÃO SEM FIO

111

As microondas viajam em linha reta, por isso, se as torres estiverem muito distantes, o a terra vai atrapalhar (pense em uma ligação entre Seattle e Amsterdã). Assim, re-os pavilhões são necessários periodicamente. Quanto mais altas as torres, mais distantes elas pode ser. A distância entre repetidores aumenta aproximadamente com a raiz quadrada da altura da torre. Para torres de 100 metros de altura, os repetidores podem estar a 80 km de distância.

Ao contrário das ondas de rádio em frequências mais baixas, as microondas não passam edifícios também. Além disso, embora o feixe possa estar bem focado no transmissor, ainda há alguma divergência no espaço. Algumas ondas podem ser refratadas fora das camadas atmosféricas baixas e pode demorar um pouco mais para chegar do que o ondas diretas. As ondas atrasadas podem chegar fora de fase com a onda direta e assim cancelar o sinal. Este efeito é chamado de **esmaecimento multipath** e é frequentemente um problema sério. Depende do clima e da frequência. Alguns operadores mantêm 10% de seus canais ociosos como sobressalentes para ligar quando o multipath enfraquece o tempo temporariamente elimina alguma banda de frequência.

A demanda por mais e mais espectro leva as operadoras a frequências ainda maiores frequências. Bandas de até 10 GHz estão agora em uso rotineiro, mas a cerca de 4 GHz um novo o problema se instala em: absorção pela água. Essas ondas têm apenas alguns centímetros longo e são absorvidos pela chuva. Este efeito seria ótimo se alguém planejasse construir um enorme forno de micro-ondas ao ar livre para assar pássaros que passam, mas para a comunidade

catão é um problema grave. Tal como acontece com o esmaecimento multipath, a única solução é desligue os links que estão sofrendo chuva e circule ao redor deles.

Em resumo, a comunicação por micro-ondas é tão amplamente usada para longa distância comunicação por telefone, telefones celulares, distribuição de televisão e outros fins afirma que se desenvolveu uma grave escassez de espectro. Ele tem várias vantagens importantes

tagens sobre a fibra. A principal delas é que não é necessário nenhum direito de passagem para estabelecer

cabos. Comprando um pequeno terreno a cada 50 km e colocando um micro-ondas torre nele, pode-se ignorar o sistema telefônico totalmente. É assim que a MCI man-envelhecida para começar como uma nova companhia telefônica de longa distância tão rapidamente. (Arrancada,

outro concorrente inicial da desregulamentada AT&T, foi completamente diferente rota: era formada pela Southern Pacific Railroad, que já possuía uma grande quantidade de direito de passagem e apenas fibra enterrada próximo aos trilhos.)

Microondas também é relativamente barato. Colocando duas torres simples (que podem ser apenas grandes postes com quatro cabos de sustentação) e colocar antenas em cada um pode ser mais barato do que enterrar 50 km de fibra em uma área urbana congestionada ou no alto de uma montanha, e também pode ser mais barato do que alugar o telefone fibra da empresa, especialmente se a companhia telefônica ainda não pagou totalmente o cobre que arrancou quando colocou a fibra.

A política do espectro eletromagnético

Para evitar o caos total, existem acordos nacionais e internacionais sobre quem pode usar quais frequências. Como todo mundo quer uma taxa de dados mais alta, todo mundo quer mais espectro. Governos nacionais alocam espectro para AM

Página 136

112

A CAMADA FÍSICA INDIVÍDUO. 2

e rádio FM, televisão e telefones celulares, bem como para companhias telefônicas, polícia, marítima, navegação, militar, governo e muitos outros concorrentes

Comercial. Em todo o mundo, uma agência da ITU-R (WRC) tenta coordenar essa alocação para que dispositivos que funcionam em vários países possam ser fabricados. No entanto, contra tentativas não estão vinculadas às recomendações do ITU-R, e o FCC (Federal Commu-Comissão de comunicação), que faz a alocação para os Estados Unidos, tem rejeitado ocasionalmente as recomendações do ITU-R (geralmente porque eles exigiam alguns grupo politicamente poderoso a desistir de alguma parte do espectro).

Mesmo quando um pedaço do espectro foi alocado para algum uso, como telefones celulares, há a questão adicional de qual operadora tem permissão para usar quais frequências. Três algoritmos foram amplamente usados no passado. O mais antigo algoritmo, muitas vezes chamado de **concurso de beleza**, exige que cada operadora explique por que seu

proposta atende melhor ao interesse público. Funcionários do governo então decidem qual das belas histórias de que mais gostam. Ter alguma proposta de prêmio oficial do governo erty no valor de bilhões de dólares para sua empresa favorita, muitas vezes leva ao suborno, corr-ruption, nepotism e pior. Além disso, mesmo um governo escrupulosamente honesto funcionário público que pensou que uma empresa estrangeira poderia fazer um trabalho melhor do que qualquer

das empresas nacionais teria muito o que explicar.

Essa observação levou ao algoritmo 2, realizando uma **loteria** entre os interessados empresas. O problema com essa ideia é que as empresas sem interesse em usar o espectro pode entrar na loteria. Se, digamos, um restaurante de fast food ou sapataria ganhar a rede, pode revender o espectro para uma operadora com um lucro enorme e sem risco. Concedendo grandes ganhos inesperados em alerta, mas de outra forma, empresas aleatórias têm sido

severamente criticado por muitos, o que levou ao algoritmo 3: **leiloar** a largura de banda para o licitante com lance mais alto. Quando o governo britânico leiloou as frequências necessária para sistemas móveis de terceira geração em 2000, esperava obter cerca de US \$ 4 bilhão. Na verdade, recebeu cerca de US \$ 40 bilhões porque as transportadoras entraram em um feed-

frenesi, morrendo de medo de perder o barco móvel. Este evento foi ativado bits gananciosos dos governos vizinhos e os inspirou a realizar seus próprios leilões. isto

funcionou, mas também deixou algumas das operadoras com tantas dívidas que estão próximas à falência. Mesmo nos melhores casos, levará muitos anos para recuperar o taxa de licenciamento.

Uma abordagem completamente diferente para alocar frequências é não alocar eles em tudo. Em vez disso, deixe todos transmitirem à vontade, mas regule a potência usada para que as estações têm um alcance tão curto que não interferem umas nas outras.

Consequentemente, a maioria dos governos reservou algumas bandas de frequência, chamadas de **Faixas ISM (Industrial, Scientific, Medical)** para uso não licenciado. Porta da garagem abridores, telefones sem fio, brinquedos controlados por rádio, mouses sem fio e vários outros dispositivos domésticos sem fio usam as bandas ISM. Para minimizar a interferência entre esses dispositivos descoordenados, a FCC determina que todos os dispositivos no As bandas ISM limitam sua potência de transmissão (por exemplo, a 1 watt) e usam outras técnicas para espalhar seus sinais em uma faixa de frequências. Os dispositivos também podem precisar de cuidado para evitar interferência nas instalações do radar.

Página 137

SEC. 2,3

TRANSMISSÃO SEM FIO

113

A localização dessas bandas varia um pouco de país para país. No Estados Unidos, por exemplo, as bandas que os dispositivos de rede usam na prática sem a necessidade de uma licença da FCC são mostradas na Figura 2-13. A banda de 900 MHz era usado para as primeiras versões do 802.11, mas está lotado. A banda de 2,4 GHz está disponível capaz na maioria dos países e amplamente utilizado para 802.11b / ge Bluetooth, embora seja sujeito à interferência de fornos de microondas e instalações de radar. 5 GHz parte do espectro inclui **U-NII (Unlicensed National Information Infraestrutura)** bandas. As bandas de 5 GHz são relativamente pouco desenvolvidas, mas, uma vez que

eles têm a maior largura de banda e são usados por 802.11a, eles estão ganhando rapidamente em popularidade.

26
MHz
902
MHz
928
MHz
2,4
GHz
5,25
GHz
5,35
GHz
5,47
GHz
5,725
GHz
Bandas U-NII
5,825
GHz
2,4835
GHz
Banda ISM
83,5
MHz
100
MHz
255
MHz
Banda ISM
100
MHz
Banda ISM

Figura 2-13. Bandas ISM e U-NII usadas nos Estados Unidos por dispositivos sem fio.

As bandas não licenciadas têm sido um grande sucesso na última década. o capacidade de usar o espectro livremente desencadeou uma grande inovação em LANs e PANs sem fio, evidenciado pela ampla implantação de tecnologia gies como 802.11 e Bluetooth. Para continuar esta inovação, mais espectro é

necessário. Um desenvolvimento empolgante nos EUA é a decisão da FCC em 2009 de permitir o uso não licenciado de **espaços em branco** em torno de 700 MHz. Os espaços em branco estão livres

bandas de frequência que foram alocadas, mas não estão sendo usadas localmente. O trans-a transição de transmissões de televisão analógica para totalmente digital nos EUA em 2010 liberou espaços em branco em torno de 700 MHz. A única dificuldade é que usar o branco espaços, os dispositivos não licenciados devem ser capazes de detectar qualquer transmitters, incluindo microfones sem fio, que têm os primeiros direitos para usar a frequência banda.

Outra onda de atividade está acontecendo em torno da banda de 60 GHz. O FCC abriu 57 GHz a 64 GHz para operação não licenciada em 2001. Esta faixa é uma enorme porção do espectro, mais do que todas as outras bandas ISM combinadas, então pode suportar o tipo de rede de alta velocidade que seria necessária para transmitir TV de alta definição no ar em sua sala de estar. A 60 GHz, rádio

Página 138

114

A CAMADA FÍSICA

INDIVÍDUO. 2

as ondas são absorvidas pelo oxigênio. Isso significa que os sinais não se propagam muito, tornando-os adequados para redes de curto alcance. As altas frequências (60 GHz) está na faixa de frequência extremamente alta ou " milímetro ", logo abaixo do infravermelho radiação) representou um desafio inicial para os fabricantes de equipamentos, mas os produtos agora são

no mercado.

2.3.4 Transmissão infravermelha

As ondas infravermelhas não guiadas são amplamente utilizadas para comunicação de curto alcance. Os controles remotos usados para televisores, videocassetes e aparelhos de som usam infravermelho. comunicação. Eles são relativamente direcionais, baratos e fáceis de construir, mas têm um principal desvantagem: eles não passam por objetos sólidos. (Tente ficar entre seu controle remoto e sua televisão e veja se ainda funciona.) Em geral, como vamos do rádio de ondas longas para a luz visível, as ondas se comportam mais e mais como luz e cada vez menos como rádio.

Por outro lado, o fato de as ondas infravermelhas não atravessarem paredes sólidas bem também é uma vantagem. Isso significa que um sistema infravermelho em uma sala de um edifício irá

não interfere com um sistema semelhante em salas ou edifícios adjacentes: você não pode controle a televisão do seu vizinho com o controle remoto. Além disso, segurança de sistemas infravermelhos contra espionagem é melhor do que a de sistemas de rádio exatamente por esse motivo. Portanto, nenhuma licença governamental é necessária para operar um sistema infravermelho, em contraste com os sistemas de rádio, que devem ser licenciados externamente

as bandas ISM. A comunicação infravermelha tem um uso limitado no desktop, por exemplo, para conectar notebooks e impressoras com o **IrDA (Infrared Data Associação)**, mas não é um jogador importante no jogo da comunicação.

2.3.5 Transmissão de luz

A sinalização óptica não guiada ou **óptica de espaço livre** tem sido usada há séculos. Paul Revere usou sinalização óptica binária da Igreja do Velho Norte, pouco antes de seu famoso passeio. Uma aplicação mais moderna é conectar as LANs em duas por meio de lasers montados em seus telhados. A sinalização óptica usando lasers é inherentemente unidirecional, então cada extremidade precisa de seu próprio laser e sua própria fotodetecção

tor. Este esquema oferece largura de banda muito alta a um custo muito baixo e é relativamente seguro porque é difícil captar um feixe de laser estreito. Também é relativamente fácil para instalar e, ao contrário da transmissão por micro-ondas, não requer uma licença da FCC. A força do laser, um feixe muito estreito, também é sua fraqueza aqui. Visando um

feixe de laser de 1 mm de largura em um alvo do tamanho de uma cabeça de alfinete a 500 metros requer

a pontaria de uma Annie Oakley moderna. Normalmente, as lentes são colocadas no sistema para desfocar ligeiramente o feixe. Para aumentar a dificuldade, vento e temperatura mudanças de estrutura podem distorcer o feixe e os feixes de laser também não podem penetrar na chuva ou neblina espessa, embora normalmente funcionem bem em dias ensolarados. No entanto, muitos de esses fatores não são um problema quando o uso é para conectar duas espaçonaves.

Página 139

SEC. 2,3

TRANSMISSÃO SEM FIO

115

Um dos autores (AST) uma vez participou de uma conferência em um hotel moderno em Europa, na qual os organizadores da conferência cuidadosamente forneceram uma sala cheia de terminais para permitir que os participantes leiam seus e-mails durante apresentações enfadonhas. Uma vez que o PTT local não estava disposto a instalar um grande número de linhas telefônicas para apenas 3 dias, os organizadores colocaram um laser no telhado e apontaram para a universidade prédio de ciência da computação a poucos quilômetros de distância. Eles testaram na noite anterior a conferência e funcionou perfeitamente. Às 9 AM em um brilhante, dia ensolarado, o link falhou completamente e permaneceu abaixado o dia todo. O padrão se repetiu no próximo dois dias. Foi só depois da conferência que os organizadores descobriram o problema: o calor do sol durante o dia causava o aumento das correntes de convecção do telhado do edifício, conforme mostrado na Fig. 2-14. Este ar turbulento desviou o feixe e o fez dançar em torno do detector, como uma estrada tremeluzente em um dia quente. A lição aqui é que trabalhar bem em condições difíceis, bem como boas condições, links ópticos não guiados precisam ser projetados com um suficiente margem de erro.

Raio Laser
sente falta do detector
Laser
Fotodetector
Região de
visão turbulenta
Aumentando o calor
fora do prédio

Figura 2-14. As correntes de convecção podem interferir no sistema de comunicação do laser tems. Um sistema bidirecional com dois lasers é mostrado aqui.

A comunicação óptica não guiada pode parecer uma tecnologia de rede exótica tecnologia hoje, mas em breve poderá se tornar muito mais prevalente. Estamos cercados

Página 140

116

A CAMADA FÍSICA

INDIVÍDUO. 2

por câmeras (que detectam a luz) e visores (que emitem luz usando LEDs e outros tecnologia). A comunicação de dados pode ser colocada em cima dessas telas por informações de codificação no padrão em que os LEDs ligam e desligam que está abaixo do limiar da percepção humana. Comunicar-se com a luz visível desta forma é inherentemente seguro e cria uma rede de baixa velocidade nas imediações do exibição. Isso poderia permitir todos os tipos de cenários de computação onipresentes fantásticos. As luzes piscando em veículos de emergência podem alertar semáforos próximos e veículos para ajudar a limpar um caminho. Sinais informativos podem transmitir mapas. Mesmo fes-Luzes ativas podem transmitir músicas sincronizadas com sua exibição.

2.4 SATÉLITES DE COMUNICAÇÃO

Na década de 1950 e início dos anos 1960, as pessoas tentaram configurar sistemas de comunicação refletindo sinais em balões meteorológicos metalizados. Infelizmente, o recebido os sinais eram muito fracos para qualquer uso prático. Então a Marinha dos Estados Unidos notou um

tipo de balão meteorológico permanente no céu - a lua - e construiu uma ópera sistema internacional para comunicação navio-terra, enviando sinais a partir dele. O progresso no campo da comunicação celestial teve que esperar até o primeiro satélite de comunicação foi lançado. A principal diferença entre um artificial satélite e um real é que o artificial pode amplificar os sinais antes enviando-os de volta, transformando uma curiosidade estranha em uma comunicação poderosa sistema.

Os satélites de comunicação têm algumas propriedades interessantes que os tornam atraente para muitas aplicações. Em sua forma mais simples, um satélite de comunicação pode ser considerado um grande repetidor de micro-ondas no céu. Contém vários **transponders**, cada um dos quais escuta alguma parte do espectro, amplifica o sinal de entrada e, em seguida, retransmite-o em outra frequência para evitar diferença com o sinal de entrada. Este modo de operação é conhecido como **dobrado tubo**. O processamento digital pode ser adicionado para manipular ou redirecionar dados separadamente

streams em toda a banda, ou informação digital pode até ser recebida pelo satélite elite e retransmissão. A regeneração de sinais dessa forma melhora o desempenho em comparação a um tubo torto porque o satélite não amplifica o ruído no sentido ascendente sinal. Os feixes descendentes podem ser largos, cobrindo uma fração substancial da superfície da Terra, ou estreita, cobrindo uma área de apenas centenas de quilômetros de diâmetro ter.

De acordo com a lei de Kepler, o período orbital de um satélite varia conforme o raio da órbita à potência 3/2. Quanto mais alto for o satélite, mais longo será o período. Perto superfície da terra, o período é de cerca de 90 minutos. Consequentemente, baixa órbita os satélites saem de vista rapidamente, por isso muitos deles são necessários para fornecer a cobertura contínua e as antenas de solo devem rastreá-los. A uma altitude de cerca de 35.800 km, o período é de 24 horas. A uma altitude de 384.000 km, o período é cerca de um mês, como qualquer pessoa que observe a lua regularmente pode testemunhar.

Página 141

SEC. 2,4

SATÉLITES DE COMUNICAÇÃO

117

O período de um satélite é importante, mas não é o único problema para determinar onde colocá-lo. Outro problema é a presença dos cintos de Van Allen, camadas de partículas altamente carregadas presas pelo campo magnético da Terra. Qualquer satélite voando dentro deles seria destruído rapidamente pelas partículas. Esses fatores levam para três regiões nas quais os satélites podem ser colocados com segurança. Essas regiões e alguns de suas propriedades são ilustradas na Figura 2-15. Abaixo, descreveremos brevemente o satélites que habitam cada uma dessas regiões.

Altitude (km)
Tipo
35.000
30.000
25.000
20.000
15.000
10.000
5.000
0
GEO
MEO
Cinto Van Allen superior
Cinto Van Allen inferior
LEO
Latência (ms)
270
35-85
1-7
Sats necessários
3
10
50

Figura 2-15. Satélites de comunicação e algumas de suas propriedades, incluindo altitude acima da terra, tempo de atraso de ida e volta e número de satélites necessários

para cobertura global.

2.4.1 Satélites geoestacionários

Em 1945, o escritor de ficção científica Arthur C. Clarke calculou que um satélite a uma altitude de 35.800 km em uma órbita equatorial circular pareceria permanecer imóvel no céu, portanto, não precisaria ser rastreado (Clarke, 1945). Ele foi para descrever um sistema de comunicação completo que utilizou esses **geo** (tripulados) **satélites estacionários**, incluindo as órbitas, painéis solares, frequências de rádio e procedimentos de lançamento. Infelizmente, ele concluiu que os satélites eram impraticáveis devido à impossibilidade de colocar amplificadores valvulados frágeis e com fome de energia em órbita, então ele nunca levou essa ideia adiante, embora tenha escrito um pouco de ciência histórias de ficção sobre isso.

A invenção do transistor mudou tudo isso, e a primeira comunicação artificial satélite de comunicação, Telstar, foi lançado em julho de 1962. Desde então, comunicação satélites se tornaram um negócio multibilionário e o único aspecto de espaço que se tornou altamente lucrativo. Esses satélites voando alto são frequentemente chamados satélites **GEO** (**Geostationary Earth Orbit**).

Página 142

118

A CAMADA FÍSICA INDIVÍDUO. 2

Com a tecnologia atual, não é aconselhável ter satélites geoestacionários espaçados muito mais perto do que 2 graus no plano equatorial de 360 graus, para evitar interferência. Com um espaçamento de 2 graus, só pode haver $360/2 = 180$ destes satélites no céu de uma vez. No entanto, cada transponder pode usar várias frequências e polarizações para aumentar a largura de banda disponível.

Para evitar o caos total no céu, a alocação de slots de órbita é feita pela ITU. Isto processo é altamente político, com países que mal saíram da idade da pedra exigindo "seus" slots de órbita (com a finalidade de alugá-los ao licitante mais alto). De outros países, no entanto, afirmam que os direitos de propriedade nacional não se estendem até o luar e que nenhum país tem o direito legal de fendas orbitais acima de seu território. Para adicionar à luta, a telecomunicação comercial não é a única aplicação. Tele emissoras de visão, governos e militares também querem um pedaço da órbita torta.

Os satélites modernos podem ser bastante grandes, pesando mais de 5000 kg e consumindo vários quilowatts de energia elétrica produzida pelos painéis solares. Os efeitos de a gravidade solar, lunar e planetária tendem a afastá-los de seus orbit slots e orientações, um efeito neutralizado por motores de foguetes de bordo. Isto A atividade de ajuste fino é chamada de **manutenção da estação**. No entanto, quando o combustível para o

motores estiver exausto (normalmente após cerca de 10 anos), o satélite deriva e cai desamparadamente, por isso tem de ser desligado. Eventualmente, a órbita decai e o O satélite entra novamente na atmosfera e queima (ou muito raramente cai na Terra).

Os slots de órbita não são o único ponto de discordia. As frequências também são um problema, porque as transmissões de downlink interferem nos usuários de micro-ondas existentes. Consequentemente, a ITU alocou certas bandas de frequência para usuários de satélite. os principais estão listados na Figura 2-16. A banda C foi a primeira a ser designada para tráfego de satélite comercial. Duas faixas de frequência são atribuídas nele, o mais baixo um para o tráfego de downlink (do satélite) e o superior para o tráfego de uplink (para o satélite). Para permitir que o tráfego vá nos dois sentidos ao mesmo tempo, dois canais são requeridos. Esses canais já estão superlotados porque também são usados por os portadores comuns para links de microondas terrestres. As bandas L e S eram adicionado por acordo internacional em 2000. No entanto, eles são estreitos e também lotado.

Banda

Downlink

Uplink

Largura de banda

Problemas
eu
1,5 GHz
1,6 GHz
15 MHz

Baixa largura de banda; lotado
S
1,9 GHz
2,2 GHz
70 MHz
Baixa largura de banda; lotado
C
4,0 GHz
6,0 GHz
500 MHz
Interferência terrestre
Ku
11 GHz
14 GHz
500 MHz
Chuva
Ka
20 GHz
30 GHz
3500 MHz
Chuva, custo do equipamento

Figura 2-16. As principais bandas de satélite.

Página 143

SEC. 2,4
SATÉLITES DE COMUNICAÇÃO

119

A próxima banda mais alta disponível para operadoras de telecomunicações comerciais é a banda Ku (K under). Esta banda (ainda) não está congestionada, e em sua maior freqüência freqüências, os satélites podem ser espaçados tão próximos quanto 1 grau. No entanto, outro problema

existe: chuva. A água absorve bem essas micro-ondas curtas. Felizmente, pesado tempestades são geralmente localizadas, portanto, usando várias estações terrestres amplamente separadas em

em vez de apenas uma, contorna o problema, mas ao preço de antenas extras, cabos e eletrônicos extras para permitir a comutação rápida entre as estações. Banda-largura também foi alocada na banda Ka (K acima) para satélite comercial tráfego, mas o equipamento necessário para usá-lo é caro. Além desses bandas comerciais, muitas bandas governamentais e militares também existem.

Um satélite moderno tem cerca de 40 transponders, na maioria das vezes com 36 MHz largura de banda. Normalmente, cada transponder opera como um tubo dobrado, mas satélites recentes

têm alguma capacidade de processamento a bordo, permitindo uma operação mais sofisticada. Nos primeiros satélites, a divisão dos transponders em canais era estática:

a largura de banda foi simplesmente dividida em bandas de frequência fixas. Hoje em dia, cada O feixe do transponder é dividido em intervalos de tempo, com vários usuários se revezando. Nós estudará essas duas técnicas (multiplexação por divisão de frequência e divisão de tempo multiplexação de íons) em detalhes posteriormente neste capítulo.

Os primeiros satélites geoestacionários tinham um único feixe espacial que iluminava cerca de 1/3 da superfície da Terra, chamada de **pegada**. Com o enorme declínio no preço, tamanho e requisitos de energia da microeletrônica, muito mais estratégia sofisticada de transmissão tornou-se possível. Cada satélite é equipado com várias antenas e vários transponders. Cada descendente o feixe pode ser focado em uma pequena área geográfica, portanto, múltiplos para cima e para baixo as transmissões da ala podem ocorrer simultaneamente. Normalmente, esses chamados **pontos os feixes** têm formato elíptico e podem ter apenas algumas centenas de km de diâmetro ter. Um satélite de comunicação para os Estados Unidos normalmente tem um feixe largo

para os 48 estados contíguos, além de feixes pontuais para o Alasca e o Havaí. Um desenvolvimento recente no mundo dos satélites de comunicação é o desenvolvimento de microestações de baixo custo, às vezes chamadas de **VSATs** (**Very Small Aperture Terminals**) (Abramson, 2000). Esses minúsculos terminais têm 1 metro ou menor tennas (versus 10 m para uma antena GEO padrão) e pode produzir cerca de 1 watt de poder. O uplink geralmente é bom para até 1 Mbps, mas o downlink costuma ser até vários megabits / s. A transmissão direta da televisão por satélite usa esta tecnologia oggi para transmissão unilateral.

Em muitos sistemas VSAT, as microestações não têm energia suficiente para comunicar-se diretamente entre si (via satélite, é claro). Em vez disso, um especial estação terrestre, o **hub**, com uma grande antena de alto ganho é necessária para retransmitir o tráfego

entre VSATs, conforme mostrado na Figura 2-17. Neste modo de operação, o emissor ou receptor tem uma grande antena e um amplificador poderoso. O trade-off é um atraso maior em troca de estações de usuário final mais baratas.

VSATs têm grande potencial em áreas rurais. Não é muito apreciado, mas mais da metade da população mundial vive a mais de uma hora de caminhada da estação mais próxima

Página 144

120

A CAMADA FÍSICA INDIVÍDUO. 2

Comunicação

satélite

1

3

2

4

Cubo

VSAT

Figura 2-17. VSATs usando um hub.

Telefone. Conectar fios de telefone a milhares de pequenas aldeias está muito além os orçamentos da maioria dos governos do Terceiro Mundo, mas instalando VSAT de 1 metro pratos alimentados por células solares costumam ser viáveis. VSATs fornecem a tecnologia que conectará o mundo.

Os satélites de comunicação têm várias propriedades que são radicalmente diferentes de links ponto-a-ponto terrestres. Para começar, embora sinais para e de um satélite que viaja à velocidade da luz (quase 300.000 km / s), o longo a distância de ida e volta introduz um atraso substancial para os satélites GEO. Dependendo na distância entre o usuário e a estação terrestre e a elevação do satélite acima do horizonte, o tempo de trânsito de ponta a ponta está entre 250 e 300 msec. Um valor típico é 270 ms (540 ms para um sistema VSAT com hub).

Para fins de comparação, os links de microondas terrestres têm uma propagação atraso de aproximadamente 3 μ sec / km, e o cabo coaxial ou links de fibra óptica têm um atraso de aproximadamente 5 μ sec / km. Os últimos são mais lentos do que os primeiros porque sinais magnéticos viajam mais rápido no ar do que em materiais sólidos.

Outra propriedade importante dos satélites é que eles são transmitidos de maneira inerente meios de comunicação. Não custa mais enviar uma mensagem a milhares de estações dentro de um pegada do transponder do que enviar para um. Para alguns aplicativos, este propriedade é muito útil. Por exemplo, pode-se imaginar uma transmissão via satélite páginas da Web populares para os caches de um grande número de computadores espalhados por um ampla área. Mesmo quando a transmissão pode ser simulada com linhas ponto a ponto,

Página 145

SEC. 2,4

SATÉLITES DE COMUNICAÇÃO

121

a transmissão por satélite pode ser muito mais barata. Por outro lado, de uma privacidade

ponto de vista, os satélites são um desastre completo: todos podem ouvir tudo. A criptografia é essencial quando a segurança é necessária. Os satélites também têm a propriedade de que o custo de transmissão de uma mensagem é dependente da distância percorrida. Uma chamada através do oceano não custa mais para atender vício do que uma chamada do outro lado da rua. Os satélites também têm excelentes taxas de erro e podem ser implantado quase que instantaneamente, uma consideração importante para resposta a desastres e milicomunicação tária.

2.4.2 Satélites de órbita terrestre média

Em altitudes muito mais baixas, entre os dois cinturões de Van Allen, encontramos o **MEO** (**Órbita Terrestre Média**) satélites. Visto da terra, eles se movem lentamente em longitude, levando cerca de 6 horas para circundar a Terra. Assim, eles devem ser rastreados conforme se movem pelo céu. Porque eles são inferiores ao GEOS, eles têm uma pegada menor no solo e requerem menos potência transmissores para alcançá-los. Atualmente, eles são usados para sistemas de navegação, em vez do que as telecomunicações, por isso não os examinaremos mais aqui. O constel-de cerca de 30 satélites **GPS** (**Sistema de Posicionamento Global**) orbitando a cerca de 20.200 km são exemplos de satélites MEO.

2.4.3 Satélites de órbita terrestre baixa

Descendo em altitude, chegamos aos satélites **LEO** (**Low-Earth Orbit**). Devido ao seu movimento rápido, um grande número deles é necessário para um sistema completo tem. Por outro lado, como os satélites estão muito próximos da terra, o solo as estações não precisam de muita energia, e o atraso de ida e volta é de apenas alguns milissegundos onds. O custo de lançamento também é substancialmente mais barato. Nesta seção, vamos examinar ine dois exemplos de constelações de satélites para serviço de voz, Iridium e Globalstar. Durante os primeiros 30 anos da era dos satélites, satélites de órbita baixa raramente eram usados porque eles entram e saem de vista tão rapidamente. Em 1990, a Motorola inovou fundamentado, preenchendo um aplicativo com a FCC pedindo permissão para lançar 77 satélites de baixa órbita para o projeto **Iridium** (o elemento 77 é irídio). O plano era posteriormente revisado para usar apenas 66 satélites, então o projeto deveria ter sido renomeado Disprósio (elemento 66), mas provavelmente soava muito como uma doença. o A ideia era que assim que um satélite saísse de vista, outro o substituiria. Essa proposta gerou um frenesi entre outras empresas de comunicação. De repente, todos queriam lançar uma cadeia de satélites de órbita baixa. Após sete anos juntando parceiros e financiamento, comunicação serviço começou em novembro de 1998. Infelizmente, a demanda comercial por grandes e pesados telefones via satélite eram insignificantes porque a rede de telefonia móvel cresceu de forma espetacular desde 1990. Como consequência, a Iridium não

122

A CAMADA FÍSICA INDIVÍDUO. 2

lucrativa e foi forçada à falência em agosto de 1999 em um dos mais fiascos corporativos espetaculares na história. Os satélites e outros ativos (no valor de \$ 5 bilhões leão) foram posteriormente comprados por um investidor por US \$ 25 milhões em uma espécie de extraterrestre venda de garagem experimental. Outros empreendimentos de negócios via satélite imediatamente seguiram o exemplo. O serviço Iridium foi reiniciado em março de 2001 e vem crescendo desde então. Ele fornece voz, dados, paging, fax e serviço de navegação em qualquer lugar em terra, ar e mar, por meio de dispositivos portáteis que se comunicam diretamente com o satélite Iridium ellites. Os clientes incluem as indústrias marítima, de aviação e de exploração de petróleo, bem como pessoas que viajam em partes do mundo sem infraestrutura de telecomunicações

(por exemplo, desertos, montanhas, o Pólo Sul e alguns países do Terceiro Mundo). Os satélites Iridium estão posicionados a uma altitude de 750 km, em polar circular órbitas. Eles são dispostos em colares norte-sul, com um satélite a cada 32 graus de latitude, conforme mostrado na Figura 2-18. Cada satélite tem no máximo 48 células (feixes de pontos) e uma capacidade de 3.840 canais, alguns dos quais são usados para paginação e navegação, enquanto outros são usados para dados e voz.

Cada satélite tem
quatro vizinhos

Figura 2-18. Os satélites Iridium formam seis colares ao redor da Terra.

Com seis colares de satélites, toda a Terra é coberta, conforme sugerido por Fig. 2-18. Uma propriedade interessante do Iridium é que a comunicação entre dis- A maioria dos clientes ocorre no espaço, como mostra a Figura 2.19 (a). Aqui vemos uma chamada er no Pólo Norte contatando um satélite diretamente acima. Cada satélite tem quatro vizinhos com os quais ele pode se comunicar, dois no mesmo colar (mostrado) e dois em colares adjacentes (não mostrados). Os satélites transmitem a chamada através deste grade até que seja finalmente enviada para o callee no Pólo Sul.

Um design alternativo ao Iridium é o **Globalstar**. É baseado em 48 satélites LEO Lite, mas usa um esquema de comutação diferente do Iridium. Considerando que Iridium retransmite chamadas de satélite para satélite, o que requer comutação sofisticada equipamento nos satélites, Globalstar usa um design tradicional de tubo dobrado. A chamada originada no Pólo Norte na Figura 2-19 (b) é enviada de volta à terra e retirada

Página 147

SEC. 2,4 SATÉLITES DE COMUNICAÇÃO

123

Tubo curvo
satélite
Chaves de satélite
no espaço
Troca
no
terra
(uma)
(b)

Figura 2-19. (a) Retransmissão no espaço. (b) Retransmissão no solo. perto da grande estação terrestre na Oficina do Papai Noel. A chamada é então encaminhada por meio de um rede terrestre para a estação terrestre mais próxima do callee e entregue por um conexão de tubo curvo conforme mostrado. A vantagem deste esquema é que ele coloca muito da complexidade no local, onde é mais fácil de gerenciar. Além disso, o uso de grandes antenas de estação terrestre que podem emitir um sinal poderoso e receber um fraco significa que telefones de baixa potência podem ser usados. Afinal, o telefone fornece apenas alguns miliwatts de energia, então o sinal que volta para o estação terrestre é bastante fraca, mesmo depois de ter sido amplificada pelo satélite. Os satélites continuam a ser lançados a uma taxa de cerca de 20 por ano, incluindo satélites cada vez maiores que agora pesam mais de 5.000 quilos. Mas também há muito pequenos satélites para a organização mais consciente do orçamento. Para tornar o espaço re-pesquisa mais acessível, acadêmicos de Cal Poly e Stanford se reuniram em 1999 para definir um padrão para satélites em miniatura e um lançador associado que reduziria muito os custos de lançamento (Nugent et al., 2008). **CubeSats** são satélites em unidades de cubos de 10 cm × 10 cm × 10 cm, cada um pesando não mais do que 1 quilograma, que podem ser lançados por apenas \$ 40.000 cada. O lançador voa como um segundo carga útil secundária em missões espaciais comerciais. É basicamente um tubo que ocupa para três unidades de cubosats e usa molas para colocá-los em órbita. Aproximadamente 20 Cubosats foram lançados até agora, com muitos mais em andamento. A maioria deles com comunicar com estações terrestres nas bandas UHF e VHF.

2.4.4 Satélites Versus Fibra

Uma comparação entre comunicação por satélite e comunicação terrestre é instrutivo. Há apenas 25 anos, pode-se argumentar que o futuro da comunicação estava com os satélites de comunicação. Afinal, o sistema telefônico

124

A CAMADA FÍSICA

INDIVÍDUO. 2

tinha mudado pouco nos 100 anos anteriores e não mostrava sinais de mudança nos próximos 100 anos. Este movimento glacial foi causado em grande parte pelo ambiente regulatório em que se esperava que as companhias telefônicas promovessem vide um bom serviço de voz a preços razoáveis (o que eles fizeram) e, em troca, lucro garantido em seu investimento. Para pessoas com dados para transmitir, 1200 bps modems estavam disponíveis. Isso era praticamente tudo o que havia.

A introdução da competição em 1984 nos Estados Unidos e um pouco mais tarde na Europa mudou tudo isso radicalmente. As companhias telefônicas começaram a substituir

suas redes de longa distância com fibra e introduziram serviços de alta largura de banda, como ADSL (Asymmetric Digital Subscriber Line). Eles também pararam seu longo tempo prática de cobrar preços artificialmente altos de usuários de longa distância para subsidiar serviço local. De repente, as conexões de fibra terrestre pareciam as vencedoras.

No entanto, os satélites de comunicação têm alguns importantes nichos de mercado que fibra não (e, às vezes, não pode) endereçar. Primeiro, quando a implantação rápida é crítica, os satélites vencem facilmente. Uma resposta rápida é útil para comunicações militares sistemas de proteção em tempos de guerra e resposta a desastres em tempos de paz. Segue o enorme terremoto de Sumatra em dezembro de 2004 e subsequente tsunami, por exemplo, os satélites de comunicação foram capazes de restaurar as comunicações ao primeiro responjas dentro de 24 horas. Esta resposta rápida foi possível porque há um de-mercado avançado de provedores de serviços de satélite no qual grandes players, como a Intelsat com mais de 50 satélites, pode alugar capacidade praticamente em qualquer lugar onde for necessária.

Para clientes atendidos por redes de satélite existentes, um VSAT pode ser configurado facilmente e rapidamente para fornecer um link megabit / s para outras partes do mundo.

Um segundo nicho é para comunicação em locais onde a infra-estrutura terrestre estrutura mal desenvolvida. Muitas pessoas hoje em dia querem se comunicar onde quer que eles vão. As redes de telefonia móvel cobrem esses locais com boas densidade populacional, mas não fazem um trabalho adequado em outros lugares (por exemplo, no mar ou em

o deserto). Por outro lado, a Iridium fornece serviço de voz em todos os lugares da Terra, mesmo no Pólo Sul. A infraestrutura terrestre também pode ser cara para instalar, dependendo do terreno e dos direitos de passagem necessários. Indonésia, por exemplo, tem seu próprio satélite para tráfego telefônico doméstico. O lançamento de um satélite foi mais barato do que amarrar milhares de cabos submarinos entre as 13.677 ilhas em o arquipélago.

Um terceiro nicho é quando a transmissão é essencial. Uma mensagem enviada por satélite pode ser recebida por milhares de estações terrestres de uma vez. Os satélites são usados para dis-homenagear grande parte da programação da TV aberta às emissoras locais por esse motivo. Há sim agora um grande mercado para transmissões via satélite de TV digital e rádio diretamente para acabar

usuários com receptores de satélite em suas casas e carros. Todo tipo de outro conteúdo também pode ser transmitido. Por exemplo, uma organização que transmite um fluxo de preços de ações, títulos ou commodities para milhares de revendedores podem encontrar um satélite sistema seja muito mais barato do que simular uma transmissão em terra.

Em suma, parece que a comunicação principal do futuro será fibra óptica restrial combinada com rádio celular, mas para alguns usos especializados,

SEC. 2,4

SATÉLITES DE COMUNICAÇÃO

125

os satélites são melhores. No entanto, há uma ressalva que se aplica a tudo isso: economia. Embora a fibra ofereça mais largura de banda, é concebível que terrestre e comunicação por satélite podem competir agressivamente no preço. Se avanços na tecnologia cortaram radicalmente o custo de implantação de um satélite (por exemplo, se algum

futuro veículo espacial pode lançar dezenas de satélites em um lançamento) ou em órbita baixa os satélites estão em alta, não é certo que a fibra ganhe todos os mercados.

2.5 MODULAÇÃO DIGITAL E MULTIPLEXAGEM

Agora que estudamos as propriedades dos canais com e sem fio, nós voltamos nossa atenção para o problema do envio de informações digitais. Fios e fios menos canais transportam sinais analógicos, como tensão variável, luz intensidade ou intensidade do som. Para enviar informações digitais, devemos conceber sinais para representar bits. O processo de conversão entre bits e sinais que representá-los é chamado **de modulação digital**.

Começaremos com esquemas que convertem diretamente bits em um sinal. Estes esquemas resultam em **transmissão de banda base**, em que o sinal ocupa frequências de zero até um máximo que depende da taxa de sinalização. É comum para fios. Em seguida, consideraremos esquemas que regulam a amplitude, fase ou frequência de um sinal portador para transmitir bits. Esses esquemas resultam em **banda passante transmissão**, em que o sinal ocupa uma banda de frequências em torno da frequência do sinal da portadora. É comum para canais sem fio e ópticos para os sinais devem residir em uma determinada banda de frequência.

Os canais geralmente são compartilhados por vários sinais. Afinal, é muito mais conveniente usar um único fio para transportar vários sinais do que instalar um fio para cada sinal. Esse tipo de compartilhamento é chamado de **multiplexação**. Isso pode ser realizado em várias maneiras diferentes. Apresentaremos métodos para tempo, frequência e código de demultiplexação de visão.

As técnicas de modulação e multiplexação que descrevemos nesta seção são todos amplamente utilizados para fios, fibra, sem fio terrestre e canais de satélite. Nas próximas seções, veremos exemplos de redes para vê-las em ação.

2.5.1 Transmissão de banda base

A forma mais direta de modulação digital é usar uma voltagem positiva para representar 1 e uma voltagem negativa para representar 0. Para uma fibra óptica, a presença de luz pode representar um 1 e a ausência de luz pode representar um 0. Este esquema é denominado **NRZ** (Non-Return-to-Zero). O nome estranho é para suas razões tóricas, e simplesmente significa que o sinal segue os dados. Um exemplo é mostrado na Figura 2-20 (b).

Uma vez enviado, o sinal NRZ se propaga pelo fio. Na outra ponta, o receptor o converte em bits por amostragem do sinal em intervalos regulares de tempo.

Página 150

126

A CAMADA FÍSICA

INDIVÍDUO. 2

(Relógio com XORed com bits)

- (a) Fluxo de bits
- (b) Não Retorno a Zero (NRZ)
- (c) Inverter NRZ (NRZI)
- (d) Manchester
- (e) codificação bipolar
(também Marca Alternativa Inversão, AMI)

1

0

0

0

1

0

1

1

1

1
Figura 2-20. Códigos de linha: (a) Bits, (b) NRZ, (c) NRZI, (d) Manchester, (e) Bipolar ou AMI.

Este sinal não será exatamente igual ao sinal enviado. Será atenuado e distorcida pelo canal e ruído no receptor. Para decodificar os bits, o receptor mapeia as amostras de sinal para os símbolos mais próximos. Para NRZ, um voltímetro será considerada para indicar que um 1 foi enviado e uma voltagem negativa será tomada para indicar que um 0 foi enviado.

NRZ é um bom ponto de partida para nossos estudos porque é simples, mas é por si só dom usado por si na prática. Esquemas mais complexos podem converter bits em sinais que atendem melhor às considerações de engenharia. Esses esquemas são chamados de **códigos de linha**.

Abaixo, descrevemos os códigos de linha que ajudam na eficiência da largura de banda, recuperação do relógio

ery e equilíbrio DC.

Eficiência de largura de banda

Com NRZ, o sinal pode alternar entre os níveis positivo e negativo para cima a cada 2 bits (no caso de 1s e 0s alternados). Isso significa que precisamos de um largura de banda de pelo menos $B / 2$ Hz quando a taxa de bits é B bits / seg. Essa relação vem da taxa de Nyquist [Eq. (2-2)]. É um limite fundamental, por isso não podemos executar NRZ mais rápido sem usar mais largura de banda. A largura de banda costuma ser um recurso limitado, mesmo

para canais com fio, os sinais de alta frequência são cada vez mais atenuados, tornando eles são menos úteis e os sinais de frequência mais alta também requerem eletrônicos mais rápidos. Uma estratégia para usar largura de banda limitada de forma mais eficiente é usar mais do que dois níveis de sinalização. Usando quatro tensões, por exemplo, podemos enviar 2 bits em uma vez como um único **símbolo**. Este projeto funcionará enquanto o sinal no receptor é suficientemente forte para distinguir os quatro níveis. A taxa em que o as mudanças de sinal são então metade da taxa de bits, portanto, a largura de banda necessária foi reduzida.

Página 151

SEC. 2,5

MODULAÇÃO DIGITAL E MULTIPLEXAGEM

127

Chamamos a taxa na qual o sinal muda a **taxa do símbolo** para distingui-lo da **taxa de bits**. A taxa de bits é a taxa de símbolo multiplicada pelo número de bits por símbolo. Um nome mais antigo para a taxa de símbolo, particularmente no contexto de serviços chamados modems telefônicos que transmitem dados digitais por linhas telefônicas, é a **taxa de transmissão**. Na literatura, os termos "taxa de bits" e "taxa de transmissão" são frequentemente usado incorretamente.

Observe que o número de níveis de sinal não precisa ser uma potência de dois.

Muitas vezes não é, com alguns dos níveis usados para proteção contra erros e simplificando o design do receptor.

Recuperação de Relógio

Para todos os esquemas que codificam bits em símbolos, o receptor deve saber quando um símbolo termina e o próximo símbolo começa a decodificar corretamente os bits. Com NRZ, em que os símbolos são simplesmente níveis de tensão, uma longa sequência de 0s ou 1s sai o sinal inalterado. Depois de um tempo, é difícil distinguir os bits, pois 15 zeros parece muito com 16 zeros, a menos que você tenha um relógio muito preciso.

Relógios precisos ajudariam com este problema, mas são uma solução cara para equipamentos commodity. Lembre-se, estamos cronometrando bits em links executados em muitos megabits / s, então o relógio teria que variar menos que uma fração de um microsegundo durante a execução mais longa permitida. Isso pode ser razoável para lento links ou mensagens curtas, mas não é uma solução geral.

Uma estratégia é enviar um sinal de clock separado para o receptor. Outro relogio

linha não é grande coisa para barramentos de computador ou cabos curtos em que existem muitos linhas em paralelo, mas é um desperdício para a maioria dos links de rede, pois se tivéssemos outro linha para enviar um sinal, poderíamos usá-lo para enviar dados. Um truque inteligente aqui é misturar o

sinal de clock com o sinal de dados por XOR os juntando de modo que nenhuma linha extra seja necessário. Os resultados são mostrados na Figura 2-20 (d). O relógio faz um relógio trans- em cada vez de bit, para que seja executado com o dobro da taxa de bits. Quando é XORed com o Nível 0, faz uma transição de baixo para alto que é simplesmente o relógio. Esta transição é um 0 lógico. Quando é XORed com o nível 1, é invertido e torna um alto para transição baixa. Esta transição é um 1 lógico. Este esquema é chamado **Manchester** codificação e foi usado para Ethernet clássico.

A desvantagem da codificação Manchester é que ela requer o dobro da banda largura como NRZ por causa do relógio, e aprendemos que largura de banda frequentemente assuntos. Uma estratégia diferente é baseada na ideia de que devemos codificar os dados para certifique-se de que há transições suficientes no sinal. Considere que NRZ irá têm problemas de recuperação de relógio apenas para execuções longas de 0s e 1s. Se houver transições frequentes, será fácil para o receptor permanecer sincronizado com o próximo fluxo de símbolos.

Como um passo na direção certa, podemos simplificar a situação codificando 1 como uma transição e um 0 como nenhuma transição, ou vice-versa. Esta codificação é chamada de **NRZI** (**Non-Return-to-Zero Inverted**), uma torção no NRZ. Um exemplo é mostrado em

Página 152

128

A CAMADA FÍSICA INDIVÍDUO. 2

Figura 2-20 (c). O popular padrão **USB** (**Universal Serial Bus**) para conexão periféricos de computador usam NRZI. Com ele, execuções longas de 1s não causam problemas. É claro que longas execuções de 0s ainda causam um problema que devemos corrigir. Se nós fossemos a companhia telefônica, podemos simplesmente exigir que o remetente não transmita também muitos 0s. As linhas de telefone digital mais antigas nos EUA, chamadas de **linhas T1**, de fato re-exija que não mais do que 15 zeros consecutivos sejam enviados para que funcionem corretamente. Para

realmente corrigir o problema, podemos quebrar execuções de 0s mapeando pequenos grupos de bits a ser transmitido de modo que grupos com 0s sucessivos sejam mapeados para um pouco mais padrões que não têm muitos 0s consecutivos.

Um código conhecido para fazer isso é chamado **4B / 5B**. Cada 4 bits é mapeado em um padrão de 5 bits com uma tabela de tradução fixa. Os cinco padrões de bits são escolhidos para que nunca haverá uma execução de mais de três 0s consecutivos. O mapeamento é mostrado na Figura 2-21. Este esquema adiciona 25% de sobrecarga, o que é melhor do que o 100% de sobrecarga da codificação Manchester. Uma vez que existem 16 combinações de entrada e 32 combinações de saída, algumas das combinações de saída não são usadas. Colocar- deixando de lado as combinações com muitos 0s sucessivos, ainda existem alguns códigos restantes. Como bônus, podemos usar esses códigos não-dados para representar a camada física

sinais de controle. Por exemplo, em alguns usos " 11111 " representa uma linha inativa e " 11000 " representa o início de um quadro.

Dados (4B) Palavra-código (5B) Dados (4B) Palavra-código (5B)

0000
11110
1000
10010
0001
01001
1001
10011
0010
10100

1010
10110
0011
10101
1011
10111
0100
01010
1100
11010
0101
01011
1101
11011
0110
01110
1110
11100
0111
01111
1111
11101

Figura 2-21. Mapeamento 4B / 5B.

Uma abordagem alternativa é fazer com que os dados pareçam aleatórios, conhecido como scrambling. Nesse caso, é muito provável que ocorram transições frequentes. UMA **scrambler** funciona por XORing os dados com uma sequência pseudo-aleatória antes que seja transmitido. Esta mistura tornará os dados tão aleatórios quanto a sequência pseudo-aleatória seqüência (assumindo que é independente da seqüência pseudo-aleatória). O receptor então XORs os bits de entrada com a mesma sequência pseudo-aleatória para recuperar os dados reais. Para que isso seja prático, a sequência pseudo-aleatória deve ser fácil de crio. É comumente fornecido como a semente para um gerador de números aleatórios simples. A codificação é atraente porque não adiciona largura de banda ou sobrecarga de tempo. No na verdade, muitas vezes ajuda a condicionar o sinal de modo que não tenha sua energia em

Página 153

SEC. 2,5

MODULAÇÃO DIGITAL E MULTIPLEXAGEM

129

componentes de frequência dominante (causados por padrões de dados repetitivos) que podem irradiam interferência eletromagnética. Scrambling ajuda porque sinais aleatórios tendem a ser " brancos " ou têm energia espalhada pelos componentes de frequência. No entanto, o embaralhamento não garante que não haverá corridas longas. Isto é possível ter azar ocasionalmente. Se os dados forem iguais aos do pseudo-aleatório seqüência, eles irão XOR para todos os 0s. Este resultado geralmente não ocorre com um seqüência pseudo-aleatória longa que é difícil de prever. No entanto, com um curto ou seqüência previsível, pode ser possível que usuários mal-intencionados enviem padrões de bits que causa longas execuções de 0s após o embaralhamento e faz com que os links falhem. Primeiras versões dos padrões para o envio de pacotes IP sobre links SONET no sistema de telefonia apresentava esse defeito (Malis e Simpson, 1999). Era possível para os usuários enviarem certos " pacotes matadores " que certamente causariam problemas.

Sinais Balanceados

Sinais que têm tanta tensão positiva quanto negativa, mesmo em curto períodos de tempo são chamados de **sinais equilibrados** . A média é zero, o que significa que eles não têm nenhum componente elétrico DC. A falta de um componente DC é uma vantagem porque alguns canais, como cabo coaxial ou linhas com transformadores, atenuam fortemente um componente DC devido às suas propriedades físicas. Além disso, um método de conectar o receptor ao canal denominado **acoplamento capacitivo** passa apenas a parte AC de um sinal. Em qualquer caso, se enviarmos um sinal cujo a média não for zero, nós desperdiçamos energia porque o componente DC será filtrado. O balanceamento ajuda a fornecer transições para a recuperação do relógio, pois há uma mistura

de tensões positivas e negativas. Ele também fornece uma maneira simples de calibrar os receptores porque a média do sinal pode ser medida e usada como uma decisão de limite para decodificar símbolos. Com sinais desequilibrados, a média pode ser drift longe do nível de decisão verdadeiro devido a uma densidade de 1s, por exemplo, que faria com que mais símbolos fossem decodificados com erros.

Uma maneira direta de construir um código balanceado é usar dois níveis de tensão para representar um 1 lógico, (digamos +1 V ou -1 V) com 0 V representando um zero. Para enviar 1, o transmissor alterna entre os níveis +1 V e -1 V, que eles sempre fazem a média. Este esquema é denominado **codificação bipolar**. Em redes telefônicas é chamado **AMI (Alternate Mark Inversion)**, baseado no antigo terminologia em que 1 é chamado de " marca " e 0 é chamado de " espaço. " Um exemplo é apresentado na Figura 2.20 (e).

A codificação bipolar adiciona um nível de voltagem para atingir o equilíbrio. Alternativamente nós podemos usar um mapeamento como 4B / 5B para alcançar o equilíbrio (bem como transições para o relógio). Um exemplo desse tipo de código balanceado é o código de linha **8B / 10B**. Isto mapeia 8 bits de entrada para 10 bits de saída, por isso é 80% eficiente, assim como o 4B / 5B código de linha. Os 8 bits são divididos em um grupo de 5 bits, que é mapeado para 6 bits, e um grupo de 3 bits, que é mapeado para 4 bits. Os símbolos de 6 e 4 bits são

Página 154

130

A CAMADA FÍSICA INDIVÍDUO. 2

em seguida, concatenado. Em cada grupo, alguns padrões de entrada podem ser mapeados para padrões de saída que têm o mesmo número de 0s e 1s. Por exemplo, " 001 " é mapeado para " 1001, " que é balanceado. Mas não há combinações suficientes para todos os padrões de saída devem ser balanceados. Para esses casos, cada padrão de entrada é mapeado

a dois padrões de saída. Um terá um 1 extra e o alternativo terá um extra 0. Por exemplo, " 000 " é mapeado para " 1011 " e seu complemento " 0100. " Conforme os bits de entrada são mapeados para os bits de saída, o codificador se lembra da **disparidade** do símbolo anterior. A disparidade é o número total de 0s ou 1s pelo qual o sinal está fora de equilíbrio. O codificador então seleciona uma saída padrão ou seu alternativo para reduzir a disparidade. Com 8B / 10B, a disparidade será no máximo 2 bits. Assim, o sinal nunca estará longe de ser equilibrado. Haverá também nunca ultrapasse cinco 1s ou 0s consecutivos, para ajudar na recuperação do relógio.

2.5.2 Transmissão de banda passante

Muitas vezes, queremos usar uma faixa de frequências que não começa em zero para enviar informações em um canal. Para canais sem fio, não é prático enviar sinais de frequência muito baixa porque o tamanho da antena precisa ser uma fração do comprimento de onda do sinal, que se torna grande. Em qualquer caso, as condições regulatórias restrições e a necessidade de evitar interferência geralmente ditam a escolha da frequências. Mesmo para fios, colocar um sinal em uma determinada banda de frequência é útil para permitir diferentes tipos de sinais coexistem no canal. Este tipo de transmissão é chamada transmissão de banda passante porque uma banda arbitrária de frequências é usada para passar o sinal.

Felizmente, nossos resultados fundamentais do início do capítulo estão todos em termos de largura de banda, ou a largura da banda de frequência. A frequência absoluta os valores não importam para a capacidade. Isso significa que podemos pegar um sinal de **banda base**

que ocupa 0 para B Hz e o desloca para cima para ocupar uma **banda passante** de S para $S + B$ Hz com- mudando a quantidade de informação que pode transportar, mesmo que o sinal ficará diferente. Para processar um sinal no receptor, podemos mudá-lo de volta para baixo à banda base, onde é mais conveniente detectar símbolos.

A modulação digital é realizada com transmissão em banda passante, regulando ou modulando um sinal de portadora que fica na banda passante. Podemos modular o amplitudde, frequênciia ou fase do sinal da portadora. Cada um desses métodos tem um cor nome de resposta. Em **ASK** (**Amplitude Shift Keying**), duas amplitudes diferentes são usados para representar 0 e 1. Um exemplo com um nível diferente de zero e zero é mostrado na Figura 2-22 (b). Mais de dois níveis podem ser usados para representar mais símbolos ols. Da mesma forma, com **FSK** (**Frequency Shift Keying**), dois ou mais tons diferentes são usados. O exemplo da Figura 2.21 (c) usa apenas duas frequências. No mais simples forma de **PSK** (**Phase Shift Keying**), a onda portadora é sistematicamente deslocada 0 ou 180 graus em cada período de símbolo. Porque existem duas fases, é chamado **BPSK** (**Binary Phase Shift Keying**). " Binário " aqui se refere aos dois símbolos, não que os símbolos representem 2 bits. Um exemplo é mostrado na Figura 2.22 (c). UMA

Página 155

SEC. 2,5
MODULAÇÃO DIGITAL E MULTIPLEXAGEM

131

melhor esquema que usa a largura de banda do canal de forma mais eficiente é usar quatro muda, por exemplo, 45, 135, 225 ou 315 graus, para transmitir 2 bits de informação por símbolo bol. Esta versão é denominada **QPSK** (**Quadrature Phase Shift Keying**).

Mudanças de fase
0
(uma)
(b)
(c)
(d)
1
0
1
1
0
0
1
0
0
1
0
0
0
0

Figura 2-22. (a) Um sinal binário. (b) Modificação de mudança de amplitude. (c) Frequênciia tecla shift. (d) Modificação de mudança de fase.

Podemos combinar esses esquemas e usar mais níveis para transmitir mais bits por símbolo. Apenas um de frequênciia e fase pode ser modulado por vez porque

eles estão relacionados, sendo a frequênciia a taxa de mudança de fase ao longo do tempo.

Normalmente, amplitude e fase são moduladas em combinação. Três exemplos são mostrado na Figura 2-23. Em cada exemplo, os pontos fornecem a amplitude legal e combinações de fases de cada símbolo. Na Fig. 2-23 (a), vemos pontos equidistantes em 45, 135, 225 e 315 graus. A fase de um ponto é indicada pelo ângulo de uma linha

dele para a origem faz com o eixo x positivo. A amplitude de um ponto é o

distância da origem. Esta figura é uma representação de QPSK.

Esse tipo de diagrama é chamado de **diagrama de constelação** . Na Fig. 2-23 (b) nós veja um esquema de modulação com uma constelação mais densa. Dezessei combinações de amplitudes e fase são usados, então o esquema de modulação pode ser usado para transmitir

Página 156

132

A CAMADA FÍSICA
INDIVÍDUO. 2

270
(uma)
90
0
180
270
(b)
90
0

270
(c)
90
0
180

Figura 2-23. (a) QPSK. (b) QAM-16. (c) QAM-64.

4 bits por símbolo. É chamado de **QAM-16**, onde QAM significa **Quadratura Am-Modulação de plitude**. A Figura 2-23 (c) é um esquema de modulação ainda mais denso com 64 combinações diferentes, portanto, 6 bits podem ser transmitidos por símbolo. É chamado **QAM-64**. Mesmo QAMs de ordem superior também são usados. Como você pode suspeitar de essas constelações, é mais fácil construir eletrônicos para produzir símbolos como uma combinação de valores em cada eixo do que como uma combinação de amplitude e fase valores. É por isso que os padrões parecem quadrados em vez de círculos concêntricos. As constelações que vimos até agora não mostram como os bits são atribuídos a símbolos. Ao fazer a atribuição, uma consideração importante é que um pequeno explosão de ruído no receptor não leva a muitos erros de bit. Isso pode acontecer se nós valores de bits consecutivos atribuídos a símbolos adjacentes. Com QAM-16, por exemplo, se um símbolo representasse 0111 e o símbolo vizinho representasse 1000, se o receptor escolhe por engano o símbolo adjacente que fará com que todos os bits sejam errado. Uma solução melhor é mapear bits para símbolos de modo que os símbolos adjacentes sejam diferentes

na posição de apenas 1 bit. Esse mapeamento é chamado de **código Gray**. Fig. 2-24 mostra um Constelação QAM-16 que foi codificada em Gray. Agora, se o receptor decodificar o símbolo em erro, ele fará apenas um único erro de bit no caso esperado de o símbolo decodificado é próximo ao símbolo transmitido.

2.5.3 Multiplexação por Divisão de Freqüência

Os esquemas de modulação que vimos nos permitem enviar um sinal para transmitir bits ao longo de um link com ou sem fio. No entanto, as economias de escala desempenham um importante

papel em como usamos as redes. Custa essencialmente a mesma quantidade de dinheiro para instalar e manter uma linha de transmissão de alta largura de banda como uma linha de baixa largura de banda

entre dois escritórios diferentes (ou seja, os custos vêm de ter que cavar a trincheira e não de que tipo de cabo ou fibra entra nele). Consequentemente, multiplexação esquemas foram desenvolvidos para compartilhar linhas entre muitos sinais.

Página 157

SEC. 2,5
MODULAÇÃO DIGITAL E MULTIPLEXAGEM

133

UMA

B

C

D

E

Quando 1101 é enviado:

Ponto

Decodifica como

Erros de bit

UMA

1101

0

B

1100

1

C

1001

1

D

1111

1

E

0101

1

1100

1000

1101

1001

```

1111
1011
1110
1010
0011
0111
0010
0110
0000
0100
0001
0101
Q
Eu

```

Figura 2-24. QAM-16 codificado em cinza.

FDM (Multiplexação por Divisão de Freqüência) tira vantagem da banda passante transmissão de compartilhar um canal. Ele divide o espectro em bandas de frequência, com cada usuário tendo a posse exclusiva de alguma banda para enviar seu sinal.

A transmissão de rádio AM ilustra o FDM. O espectro alocado é de cerca de 1 MHz, aproximadamente 500 a 1500 kHz. Diferentes frequências são alocadas para diferentes lógicas canais (estações), cada um operando em uma parte do espectro, com a separação entre canais grande o suficiente para evitar interferência.

Para um exemplo mais detalhado, na Figura 2-25, mostramos três tele-voz canais de telefone multiplexados usando FDM. Os filtros limitam a largura de banda utilizável a cerca de 3100 Hz por canal de grau de voz. Quando muitos canais são multiplexados juntos, 4000 Hz são alocados por canal. O excesso é chamado de **banda de guarda**. Isto mantém os canais bem separados. Primeiro, os canais de voz são aumentados em frequência, cada um por uma quantidade diferente. Então eles podem ser combinados porque não há dois canais

nels agora ocupam a mesma porção do espectro. Observe que embora haja existem lacunas entre os canais graças às bandas de guarda, há alguma sobreposição entre canais adjacentes. A sobreposição existe porque os filtros reais não têm bordas afiadas ideais. Isso significa que um pico forte na borda de um canal é sentido no adjacente como ruído não térmico.

Este esquema tem sido usado para multiplexar chamadas no sistema telefônico para muitos anos, mas agora é preferível a multiplexação no tempo. No entanto, FDM continua a ser usado em redes de telefonia, bem como celular, sem fio terrestre, e redes de satélite em um nível mais alto de granularidade.

Ao enviar dados digitais, é possível dividir o espectro de forma eficiente sem usar bandas de guarda. Em **OFDM (Divisão de Frequência Ortogonal Multiplexing)**, a largura de banda do canal é dividida em muitas subportadoras que independentemente enviar dados imediatamente (por exemplo, com QAM). As subportadoras são embaladas firmemente juntas em

o domínio da frequência. Assim, os sinais de cada subportadora se estendem para os uns. No entanto, como visto na Fig. 2-26, a resposta de frequência de cada subportadora é

134

A CAMADA FÍSICA INDIVÍDUO. 2

```

300
3100
Canal 3
Canal 2
Canal 1
1
1
1
Atenuação
fator
64
Frequênci(a) (kHz)
(c)
Canal 1
Canal 3
Canal 2
68
72

```

60
64
Frequência (kHz)
(b)
Frequência (Hz)
(uma)
68
72
60

Figura 2-25. Multiplexação por divisão de frequência. (a) As larguras de banda originais.

(b) As larguras de banda aumentaram em frequência. (c) O canal multiplexado.

projetado de modo que seja zero no centro das subportadoras adjacentes. As subportadoras podem, portanto, ser amostrados em suas frequências centrais sem interferência de seus vizinhos. Para fazer este trabalho, um tempo de guarda é necessário para repetir uma parte do símbolo sinaliza a tempo para que tenham a resposta de frequência desejada.

No entanto, esse overhead é muito menor do que o necessário para muitas bandas de guarda.

Frequência
Poder
f_3
f_4
f_2
f_1
f_5
Separação
f
Uma subportadora OFDM (sombreada)

Figura 2-26. Multiplexação por divisão ortogonal de frequência (OFDM).

A ideia de OFDM já existe há muito tempo, mas é apenas no último década que vem sendo amplamente adotada, a partir da constatação de que é possível

Página 159

SEC. 2,5

MODULAÇÃO DIGITAL E MULTIPLEXAGEM

135

para implementar OFDM de forma eficiente em termos de uma transformação de Fourier de dados digitais

sobre todas as subportadoras (em vez de modular separadamente cada subportadora). OFDM é usado em 802.11, redes de cabo e redes de linha de energia, e está planejado para sistemas celulares de quarta geração. Normalmente, um fluxo de alta taxa de informações digitais mação é dividida em muitos fluxos de baixa taxa que são transmitidos nas subportadoras em paralelo. Esta divisão é valiosa porque as degradações do canal são fáceis er para lidar com o nível da subportadora; algumas subportadoras podem estar muito degradadas e excluídos em favor de subportadoras que são bem recebidas.

2.5.4 Multiplexação por Divisão de Tempo

Uma alternativa ao FDM é o **TDM** (Time Division Multiplexing). Aqui os usuários se revezam (no modo round-robin), cada um obtendo periodicamente o largura de banda por um pequeno intervalo de tempo. Um exemplo de três fluxos sendo multiplexed com TDM é mostrado na Figura 2-27. Bits de cada fluxo de entrada são coletados um **intervalo de tempo** fixo e saída para o fluxo agregado. Este fluxo corre na soma taxa dos fluxos individuais. Para que isso funcione, os fluxos devem ser sincronizados em tempo. Pequenos intervalos de tempo de **guarda** análogos a uma banda de guarda de frequência podem ser adicionado para acomodar pequenas variações de tempo.

1
2
3
Round-robin
TDM
multiplexador
3
2
3
1
2
1
Tempo de guarda
2

Figura 2-27. Multiplexação por divisão de tempo (TDM).

O TDM é amplamente usado como parte das redes de telefone e celular. Evitar

um ponto de confusão, vamos deixar claro que é bastante diferente da alternativa **STDM** (**Multiplexação por divisão de tempo estatística**). O prefixo " estatístico " é adicionado para indicar que os fluxos individuais contribuem para o fluxo multiplexado *não* em um horário fixo, mas de acordo com as estatísticas de sua demanda. STDM é comutação de pacotes por outro nome.

2.5.5 Multiplexação por divisão de código

Existe um terceiro tipo de multiplexação que funciona de uma maneira completamente diferente do que FDM e TDM. **CDM** (**Code Division Multiplexing**) é uma forma de **propagação** comunicação de **espectro** em que um sinal de banda estreita é espalhado por um banda de frequência mais ampla. Isso pode torná-lo mais tolerante a interferências, bem como permitindo que vários sinais de diferentes usuários compartilhem a mesma banda de frequência. Porque a multiplexação por divisão de código é usada principalmente para o último propósito, é com- normalmente denominado **CDMA** (**Code Division Multiple Access**).

Página 160

136

A CAMADA FÍSICA INDIVÍDUO. 2

O CDMA permite que cada estação transmita em todo o espectro de frequência, todos A Hora. Múltiplas transmissões simultâneas são separadas usando a teoria da codificação. Antes de entrar no algoritmo, vamos considerar uma analogia: um saguão de aeroporto com muitos pares de pessoas conversando. TDM é comparável a pares de pessoas em a sala se revezando para falar. FDM é comparável aos pares de pessoas que falam - em tons diferentes, alguns agudos e outros graves, de modo que cada par pode manter sua própria conversa ao mesmo tempo, mas independentemente das outras. O CDMA é comparável a cada par de pessoas falando ao mesmo tempo, mas de uma forma diferente linguagem diferente. O casal de língua francesa apenas se concentra no francês, rejeita transformando tudo o que não é francês como ruído. Assim, a chave para CDMA é ser capaz de extraia o sinal desejado enquanto rejeita todo o resto como ruído aleatório. UMA segue uma descrição um tanto simplificada do CDMA.

No CDMA, cada tempo de bit é subdividido em m intervalos curtos chamados **chips** . Normalmente, existem 64 ou 128 chips por bit, mas no exemplo dado aqui, iremos usar 8 chips / bit para simplificar. Cada estação recebe um código m - bit exclusivo chamado uma **sequência de chips** . Para fins pedagógicos, é conveniente usar uma nota bipolar para escrever esses códigos como sequências de -1 e +1. Vamos mostrar o chip se-referências entre parênteses.

Para transmitir um bit 1, uma estação envia sua sequência de chips. Para transmitir um bit 0, envia a negação de sua sequência de chips. Nenhum outro padrão é permitido. Portanto, para $m = 8$, se a estação *A* for atribuída à sequência de chips $(-1 \ -1 \ -1 \ +1 \ +1 \ -1 \ +1 \ +1)$, pode enviar 1 bit transmitindo a sequência do chip e 0 transmitindo $(+1 \ +1 \ +1 \ -1 \ -1 \ +1 \ -1 \ -1)$. É realmente sinais com esses níveis de tensão que são enviados, mas é suficiente pensarmos em termos das sequências.

Aumentando a quantidade de informações a serem enviadas de b bits / seg para MB chips / s para cada estação significa que a largura de banda necessária para CDMA é maior por um fator de m do que a largura de banda necessária para uma estação que não usa CDMA (assumindo

sem alterações nas técnicas de modulação ou codificação). Se tivermos um 1 MHz banda disponível para 100 estações, com FDM cada uma teria 10 kHz e poderia enviar a 10 kbps (assumindo 1 bit por Hz). Com CDMA, cada estação usa o 1 completo MHz, então a taxa de chip é de 100 chips por bit para espalhar a taxa de bits da estação de 10 kbps através do canal.

Na Figura 2-28 (a) e (b), mostramos as sequências de chips atribuídas a quatro exemplos estações e os sinais que representam. Cada estação tem seu próprio chip exclusivo seqüência. Vamos usar o símbolo **S** para indicar o vetor m - chip para a estação *S* , e **S** para sua negação. Todas as sequências de chips são **ortogonais aos pares** , pelo que nós

significa que o produto interno normalizado de quaisquer duas sequências de chips distintas, S e T (escrito como ST), é 0. Sabe-se como gerar tal chip ortogonal se-respostas usando um método conhecido como **códigos de Walsh**. Em termos matemáticos, ortogonalidade das sequências de chips pode ser expressa da seguinte forma:

$$ST \equiv$$

$$\sum_{i=1}^m S_i T_i = 0$$

(2-5)

Página 161

SEC. 2,5

MODULAÇÃO DIGITAL E MULTIPLEXAGEM

137

Em linguagem simples, tantos pares são iguais quanto diferentes. Esta ortogonalidade propriedade será crucial mais tarde. Observe que se $ST = 0$, então ST também é 0. O produto interno normalizado de qualquer sequência de chips com ele mesmo é 1:

$$SS =$$

$$\sum_{i=1}^m S_i S_i =$$

$$\sum_{i=1}^m (\pm 1)^2 = 1$$

Isso ocorre porque cada um dos m termos no produto interno é 1, então a soma é m . Observe também que $SS = -1$.

(b)

$$\begin{aligned} A &= (-1 -1 -1 +1 +1 -1 +1 +1) \\ B &= (-1 -1 +1 -1 +1 +1 +1 -1) \\ C &= (-1 +1 -1 +1 +1 +1 -1 -1) \\ D &= (-1 +1 -1 -1 -1 +1 -1) \\ &\text{(uma)} \\ &\text{(c)} \\ &\text{(d)} \\ S_1 &= C \\ &= (-1 +1 -1 +1 +1 +1 -1 -1) \\ S_2 &= B + C \\ &= (-2 0 0 0 +2 +2 0 -2) \\ S_3 &= A + B \\ &= (0 0 -2 +2 0 -2 0 +2) \\ S_4 &= A + B + C \\ &= (-1 +1 -3 +3 +1 -1 -1 +1) \\ S_5 &= A + B + C + D = (-4 0 -2 0 +2 0 +2 -2) \\ S_6 &= A + B + C + D = (-2 -2 0 -2 0 -2 +4 0) \\ S_1 C &= [1 + 1 -1 + 1 + 1 + 1 -1 -1] / 8 = 1 \\ S_2 C &= [2 + 0 + 0 + 0 + 2 + 2 + 0 + 2] / 8 = 1 \\ S_3 C &= [0 + 0 + 2 + 2 + 0 -2 + 0 -2] / 8 = 0 \\ S_4 C &= [1 + 1 + 3 + 3 + 1 -1 + 1 -1] / 8 = 1 \\ S_5 C &= [4 + 0 + 2 + 0 + 2 + 0 -2 + 2] / 8 = 1 \end{aligned}$$

$$S_6 C = [2-2 + 0-2 + 0-2-4 + 0] / 8 = -1$$

Figura 2-28. (a) Sequências de chips para quatro estações. (b) Sinaliza as sequências representam (c) Seis exemplos de transmissões. (d) Recuperação do sinal da estação C.

Durante cada tempo de bit, uma estação pode transmitir um 1 (enviando sua sequência de chip), ele pode transmitir um 0 (enviando o negativo de sua sequência de chips) ou pode ser silencioso e não transmitir nada. Assumimos por agora que todas as estações estão sincronizadas no tempo, então todas as sequências de chips começam no mesmo instante. Quando duas ou mais esta-

cões transmitem simultaneamente, suas sequências bipolares somam linearmente. Por exemplo, se em um período de chip três estações emitem +1 e uma estação produz -1, +2 ser recebido. Pode-se pensar nisso como sinais que somam tensões superpostas o canal: três estações de saída +1 V e uma estação de saídas -1 V, de modo que 2 V é recebido. Por exemplo, na Figura 2-28 (c), vemos seis exemplos de um ou mais estados transmitindo 1 bit ao mesmo tempo. No primeiro exemplo, C transmite um bit 1, então nós apenas obtemos a sequência de chips de C. No segundo exemplo, B e C transmitem 1 bits, então obtemos a soma de suas sequências de chips bipolares, a saber:

$$(-1 -1 +1 -1 +1 +1 -1) + (-1 +1 -1 +1 +1 -1 -1) = (-2 \ 0 \ 0 \ 0 +2 +2 \ 0 \ -2)$$

Para recuperar o fluxo de bits de uma estação individual, o receptor deve saber que sequência de chips da estação com antecedência. Ele faz a recuperação calculando o produto interno malizado da sequência de chips recebida e a sequência de chips do estação cujo fluxo de bits está tentando recuperar. Se a sequência de chips recebida for S e o receptor está tentando ouvir uma estação cuja sequência de chips é C , apenas calcula o produto interno normalizado, SC .

138

A CAMADA FÍSICA INDIVÍDUO. 2

Para ver por que isso funciona, imagine que duas estações, A e C, transmitem uma 1 bit ao mesmo tempo que B transmite o bit 0, como no terceiro exemplo.

O receptor vê a soma, $S = A + B + C$, e calcula

$$SC = (A + B + C)C = AC + BC + CC = 0 + 0 + 1 = 1$$

Os primeiros dois termos desaparecem porque todos os pares de sequências de chips foram cuidadosamente

escolhido para ser ortogonal, conforme mostrado na Eq. (2-5). Agora deve ficar claro por que isso a propriedade deve ser imposta às sequências de chips.

Para tornar o processo de decodificação mais concreto, mostramos seis exemplos em Figura 2-28 (d). Suponha que o receptor esteja interessado em extrair o bit enviado por estação C de cada um dos seis sinais S_1 a S_6 . Calcula bit por soma combinar os produtos dos pares dos vetores S e C recebidos da Figura 2-28 (a) e em seguida, tomando 1/8 do resultado (já que $m = 8$ aqui). Os exemplos incluem casos onde C é silencioso, envia um bit 1 e envia um bit 0, individualmente e em combinação com outras transmissões. Conforme mostrado, o bit correto é decodificado a cada vez. Isto é assim como falar francês.

Em princípio, dada a capacidade de computação suficiente, o receptor pode ouvir todos os remetentes de uma vez, executando o algoritmo de decodificação para cada um deles em paralelo. Na vida real, basta dizer que isso é mais fácil dizer do que fazer, e é útil saber quais remetentes podem estar transmitindo.

No sistema CDMA ideal e silencioso que estudamos aqui, o número de sequências que enviam simultaneamente podem ser feitas arbitrariamente grandes usando séries de chips mais longas

quências. Para 2^n estações, os códigos Walsh podem fornecer 2^n sequências de chips ortogonais de comprimento 2^n . No entanto, uma limitação significativa é que assumimos que todos os chips são sincronizados no tempo no receptor. Esta sincronização não é até mesmo aproximadamente verdadeiro em algumas aplicações, como redes celulares (nas quais CDMA foi amplamente implantado a partir da década de 1990). Isso leva a diferentes de-sinais. Retornaremos a este tópico mais tarde no capítulo e descreveremos como assincronous CDMA difere do CDMA síncrono.

Assim como nas redes celulares, o CDMA é usado por satélites e redes de cabo. Abordamos muitos fatores complicadores nesta breve introdução. En-gineers que desejam obter uma compreensão profunda do CDMA devem ler Viterbi (1995) e Lee e Miller (1998). Essas referências exigem um pouco de retorno base em engenharia de comunicação, no entanto.

2.6 A REDE TELEFÔNICA PÚBLICA COMUTADA

Quando dois computadores pertencentes à mesma empresa ou organização e localização próximos uns dos outros precisam se comunicar, muitas vezes é mais fácil apenas passar um cabo entre eles. LANs funcionam dessa maneira. No entanto, quando as distâncias são grandes ou existem muitos computadores ou os cabos têm que passar por uma via pública ou outro direito de passagem público, os custos de execução de cabos privados são geralmente proibitivos.

Página 163

SEC. 2,6

A REDE PÚBLICA DE TELEFONE MUDADA

139

Além disso, em quase todos os países do mundo, amarrando transmissões privadas linhas de ação cruzando (ou abaixo) da propriedade pública também é ilegal. Consequentemente, os projetistas de redes devem contar com as instalações de telecomunicações existentes.

Essas facilidades, principalmente a **PSTN** (**Rede Telefônica Pública Comutada - trabalho**), geralmente eram projetados há muitos anos, com um objetivo completamente diferente

em mente: transmitir a voz humana de uma forma mais ou menos reconhecível. Seus a adequação para uso em comunicação computador-computador é freqüentemente marginal, na melhor das hipóteses.

Para ver o tamanho do problema, considere que um cabo de commodity barato rodando entre dois computadores podem transferir dados a 1 Gbps ou mais. Em contraste, típico ADSL, a alternativa incrivelmente rápida para um modem de telefone, funciona em cerca de 1 Mbps. A diferença entre os dois é a diferença entre o cruzeiro em um avião e dar um passeio.

No entanto, o sistema telefônico está fortemente interligado com (área ampla) redes de computadores, por isso vale a pena dedicar algum tempo para estudá-lo em detalhes. o fator limitante para fins de rede acaba sendo a " última milha " sobre a qual os clientes se conectam, não os troncos e switches dentro da rede telefônica.

Esta situação está mudando com o lançamento gradual da fibra e da tecnologia digital na borda da rede, mas vai levar tempo e dinheiro. Durante a longa espera, designers de sistemas de computador acostumados a trabalhar com sistemas que dão pelo menos três ordens de magnitude melhor desempenho dedicaram muito tempo e esforço para descobrir

Saiba como usar a rede telefônica com eficiência.

Nas seções a seguir, descreveremos o sistema telefônico e mostraremos como funciona. Para obter informações adicionais sobre as entradas do sistema telefônico, consulte Bellamy (2000).

2.6.1 Estrutura do Sistema Telefônico

Logo depois que Alexander Graham Bell patenteou o telefone em 1876 (apenas alguns horas antes de seu rival, Elisha Gray), havia uma enorme demanda por seu novo invenção. O mercado inicial era para a venda de telefones, que chegavam aos pares.

Cabia ao cliente colocar um único fio entre eles. Se um proprietário queria falar com n outros proprietários de telefone, fios separados tiveram que ser instalados

para todas as n casas. Em um ano, as cidades estavam cobertas de fios passando por casas e árvores em uma confusão selvagem. Tornou-se imediatamente óbvio que o modelo de conectar todos os telefones a todos os outros telefones, conforme mostrado na Figura 2-29 (a), não estava indo funcionar.

Para seu crédito, Bell percebeu esse problema no início e formou a Bell Telephone Company, que abriu seu primeiro escritório de comutação (em New Haven, Connecticut) em 1878. A empresa fez uma transferência para a casa ou escritório de cada cliente. Fazer um chamada, o cliente ligaria o telefone para fazer um som de toque no telefone

escritório da empresa para atrair a atenção de um operador, que então manualmente conecte o chamador ao receptor usando um cabo curto para conectar o chamador para o receptor. O modelo de uma única central de comutação é ilustrado na Figura 2.29 (b).

Página 164

140

A CAMADA FÍSICA

INDIVÍDUO. 2

- (uma)
- (b)
- (c)

Figura 2-29. (a) Rede totalmente interconectada. (b) Chave centralizada.

(c) Hierarquia de dois níveis.

Em breve, escritórios de comutação da Bell System surgiram em todos os lugares e as pessoas queriam fazer chamadas de longa distância entre cidades, então o Sistema Bell começou a conectar os escritórios de comutação. O problema original logo voltou: para conectar todas as centrais de comutação a todas as outras centrais de comutação por meio de um fio entre eles rapidamente se tornou incontrolável, então escritórios de comutação de segundo nível foram inventados. Depois de um tempo, vários escritórios de segundo nível foram necessários, como ilustrado

tratado na Figura 2.29 (c). Eventualmente, a hierarquia cresceu para cinco níveis.

Em 1890, as três partes principais do sistema telefônico já existiam: os escritórios de comutação, os fios entre os clientes e os escritórios de comutação (por agora pares equilibrados, isolados e trançados em vez de fios abertos com retorno à terra), e as conexões de longa distância entre as centrais de comutação. Por um curto

história técnica do sistema telefônico, ver Hawley (1991).

Embora tenha havido melhorias em todas as três áreas desde então, o básico

O modelo do Bell System permaneceu essencialmente intacto por mais de 100 anos. O seguinte A descrição a seguir é altamente simplificada, mas fornece o sabor essencial.

Cada telefone tem dois fios de cobre que saem dele e vão diretamente para o telefone estação **final** mais próxima da companhia telefônica (também chamada de **escritório central local**). O dis-

A distância é normalmente de 1 a 10 km, sendo mais curta nas cidades do que nas áreas rurais. No Só nos Estados Unidos, existem cerca de 22.000 estações finais. As conexões de dois fios entre o telefone de cada assinante e a estação final são conhecidos no comércio como o **loop local**. Se os loops locais do mundo fossem estendidos de ponta a ponta, eles iria se estender até a lua e voltar 1000 vezes.

Ao mesmo tempo, 80% do valor de capital da AT&T era o cobre nos loops locais.

A AT&T era, então, a maior mina de cobre do mundo. Felizmente, este fato não era muito conhecido na comunidade de investidores. Se fosse conhecido, alguns cor-
Porate raider pode ter comprado a AT&T, encerrado todo o serviço telefônico nos Estados Unidos Estados Unidos, arrancou todo o fio e vendeu-o a um refinador de cobre para um retorno rápido.

Página 165

SEC. 2,6

A REDE PÚBLICA DE TELEFONE MUDADA

141

Se um assinante conectado a uma determinada estação final ligar para outro assinante conectado para a mesma estação final, o mecanismo de comutação dentro do escritório configura um direto conexão elétrica entre os dois loops locais. Esta conexão permanece intacta durante a chamada.

Se o telefone chamado estiver conectado a outra estação final, um procedimento diferente tem que ser usado. Cada estação final tem várias linhas de saída para um ou mais centros de comutação próximos, chamados **postos de pedágio** (ou, se eles estiverem no mesmo local área, **escritórios tandem**). Essas linhas são chamadas de **troncos de conexão interurbana**. O número de diferentes tipos de centros de comutação e sua topologia varia de país para o país dependendo da densidade telefônica do país.

Se as estações finais do chamador e do receptor tiverem uma ligação tarifada tronco para o mesmo posto de pedágio (uma ocorrência provável se eles estiverem relativamente próximos), a conexão pode ser estabelecida dentro da central de pedágio. Uma rede telefônica consistindo apenas em telefones (os pontos pequenos), estações finais (os pontos grandes) e pedágio escritórios (os quadrados) é mostrado na Figura 2.29 (c). Se o chamador e o receptor não tiverem uma central de pedágio em comum, um caminho terá que ser estabelecido entre duas agências de pedágio. As agências de pedágio se comunicam com cada outro via **troncos intertoll** de alta largura de banda (também chamados de **troncos interoffice**). Anterior à dissolução da AT&T em 1984, o sistema de telefonia dos Estados Unidos usava uma rota hierárquica para encontrar um caminho, indo para níveis mais altos da hierarquia até que houvesse uma mudança escritório em comum. Isso foi então substituído por mais flexível, não hierárquico roteamento. A Figura 2-30 mostra como uma conexão de longa distância pode ser roteada.

Telefone
Fim
escritório
Pedágio
escritório
Intermediário
trocando
escritório (s)
Telefone
Fim
escritório
Pedágio
escritório
Local
ciclo
Pedágio
conectando
tronco
Muito alto
largura de banda
intertoll
roupa de baixo
Pedágio
conectando
tronco
Local
ciclo

Figura 2-30. Uma rota de circuito típica para uma chamada de longa distância.

Uma variedade de meios de transmissão são usados para telecomunicações. Ao contrário edifícios de escritórios modernos, onde a fiação é comumente Categoria 5, loops locais para casas consistem principalmente de pares trançados de categoria 3, com fibra apenas começando a aparecer. Entre escritórios de comutação, cabos coaxiais, microondas e especialmente as fibras ópticas são amplamente utilizadas.

No passado, a transmissão em todo o sistema telefônico era analógica, com o sinal de voz real sendo transmitido como uma voltagem elétrica da fonte para destino. Com o advento da fibra óptica, eletrônica digital e computadores, todos os troncos e interruptores agora são digitais, deixando o loop local como a última parte do

tecnologia analógica no sistema. A transmissão digital é preferida porque é não é necessário reproduzir com precisão uma forma de onda analógica depois que ela passou através de muitos amplificadores em uma chamada longa. Ser capaz de distinguir corretamente um 0 de 1 é suficiente. Esta propriedade torna a transmissão digital mais confiável do que analógico. Também é mais barato e fácil de manter.

Em resumo, o sistema telefônico consiste em três componentes principais:

1. Loops locais (pares trançados analógicos que vão para casas e empresas).
2. Troncos (links de fibra óptica digital conectando as centrais de comutação).
3. Comutação de escritórios (onde as chamadas são movidas de um tronco para outro).

Depois de uma curta digressão sobre a política dos telefones, voltaremos a cada desses três componentes com algum detalhe. Os loops locais fornecem acesso a todos cesso a todo o sistema, então eles são críticos. Infelizmente, eles também são elo mais fraco do sistema. Para os troncos de longa distância, a questão principal é como coletar Faça várias chamadas em conjunto e envie-as pela mesma fibra. Isso exige multiplexação, e aplicamos FDM e TDM para fazê-lo. Finalmente, existem dois fundamentos maneiras mentalmente diferentes de fazer a comutação; vamos olhar para ambos.

2.6.2 A Política dos Telefones

Por décadas antes de 1984, o Sistema Bell fornecia tanto locais quanto de longa distância serviço de assistência na maior parte dos Estados Unidos. Na década de 1970, o Federal US O governo passou a acreditar que este era um monopólio ilegal e processou para quebrar isso. O governo venceu e, em 1º de janeiro de 1984, a AT&T foi dividida em AT&T Long Lines, 23 BOCs (**Bell Operating Companies**) e algumas outras peças. Os 23 BOCs foram agrupados em sete BOCs regionais (RBOCs) para fazer eles são economicamente viáveis. Toda a natureza das telecomunicações nos Estados Unidos Os estados foram alterados da noite para o dia por ordem do tribunal (*não* por um ato do Congresso).

As especificações exatas da alienação foram descritas nos chamados

MFJ (Julgamento Final Modificado), um oxímoro, se já houve um - se o o julgamento podia ser modificado, obviamente não era final. Este evento levou a um aumento concorrência, melhor serviço e menores tarifas de longa distância para consumidores e empresas esses. No entanto, os preços do serviço local aumentaram à medida que os subsídios cruzados de longa data

as ligações à distância foram eliminadas e o serviço local passou a ser autossustentável.

Muitos outros países já introduziram concorrência em linhas semelhantes.

De relevância direta para nossos estudos é que o novo quadro competitivo fez com que um recurso técnico importante fosse adicionado à arquitetura da rede telefônica trabalhos. Para deixar claro quem poderia fazer o quê, os Estados Unidos foram divididos em 164 LATA (**Áreas de Acesso e Transporte Local**). Grosso modo, uma LATA é quase tão grande quanto a área coberta por um código de área. Dentro de cada LATA, havia um LEC (**Local Exchange Carrier**) com monopólio do telefone tradicional

Página 167

SEC. 2,6

A REDE PÚBLICA DE TELEFONE MUDADA

143

serviço dentro de sua área. Os LECs mais importantes foram os BOCs, embora alguns LATA continham uma ou mais das 1.500 companhias telefônicas independentes operando como LECs.

O novo recurso era que todo o tráfego inter-LATA era tratado por um diferente tipo de empresa, um IXC (**IntereXchange Carrier**). Originalmente, AT&T Long Lines era o único IXC sério, mas agora existem concorrentes bem estabelecidos como Verizon e Sprint no negócio IXC. Uma das preocupações no separação era para garantir que todos os IXCs seriam tratados igualmente em termos de linha qualidade, tarifas e o número de dígitos que seus clientes teriam que discar para usar eles. A maneira como isso é tratado é ilustrada na Figura 2-31. Aqui vemos três ex-LATAs amplas, cada uma com várias estações finais. LATAs 2 e 3 também têm um pequeno hierarquia com escritórios tandem (postos de pedágio intra-LATA).

1

2

Para loops locais

IXC # 1's

pedágio

IXC # 2's

pedágio

IXC POP

Tandem

escritório

Fim

escritório

LATA 3

LATA 2
LATA 1
1
2
1
2
1
2

Figura 2-31. A relação de LATAs, LECs e IXCs. Todos os círculos são

Escrítorios de comutação LEC. Cada hexágono pertence ao IXC cujo número está nele.

Qualquer IXC que deseja atender chamadas originadas em uma LATA pode construir um escritório de comutação chamado **POP (Ponto de Presença)** lá. O LEC é necessário para conectar cada IXC a cada estação final, seja diretamente, como em LATAs 1 e 3, ou indiretamente, como em LATA 2. Além disso, os termos da conexão, tanto técnico como financeiro, devem ser idênticos para todos os IXCs. Este requisito permite, um subescriba em, digamos, LATA 1, para escolher qual IXC usar para ligar para assinantes em LATA 3.

Como parte do MFJ, os IXCs foram proibidos de oferecer serviço telefônico local e os LECs foram proibidos de oferecer serviço telefônico inter-LATA, embora

Página 168

144

A CAMADA FÍSICA

INDIVÍDUO. 2

ambos estavam livres para entrar em qualquer outro negócio, como operar restaurantes de frango frito

rants. Em 1984, essa foi uma declaração bastante inequívoca. Infelizmente, tecnologia Gy tem uma maneira engraçada de tornar a lei obsoleta. Nem televisão a cabo nem móveis telefones biliares foram cobertos pelo acordo. Como a televisão a cabo passou de um caminho de mão dupla e telefones celulares explodiram em popularidade, tanto LECs quanto IXCs começou a comprar ou se fundir com operadoras de cabo e móvel.

Em 1995, o Congresso viu que tentar manter uma distinção entre as nossos tipos de empresas não eram mais sustentáveis e elaborou um projeto de lei para preservar possibilidade de concorrência, mas permitir empresas de TV a cabo, telefone local empresas, operadoras de longa distância e operadoras de telefonia móvel para entrar no negócio um do outro

esses. A ideia era que qualquer empresa poderia oferecer a seus clientes um único pacote integrado contendo TV a cabo, telefone e serviços de informação e que diferentes empresas competiriam em serviço e preço. A conta foi então transformou-se em lei em fevereiro de 1996 como uma grande revisão do regulamento de telecomunicações

ção. Como resultado, alguns BOCs tornaram-se IXCs e algumas outras empresas, como operadoras de televisão a cabo, começaram a oferecer serviço de telefonia local em competição com os LECs.

Uma propriedade interessante da lei de 1996 é a exigência de que os LECs implementem portabilidade do número local . Isso significa que um cliente pode mudar de local companhias telefônicas sem ter que obter um novo número de telefone. Portabilidade para números de telefone celular (e entre linhas fixas e móveis) seguiram o exemplo em 2003. Essas disposições removeram um grande obstáculo para muitas pessoas, tornando-as muito mais inclinado a trocar LECs. Como resultado, as telecomunicações dos EUA paisagem tornou-se muito mais competitiva, e outros países seguiram o exemplo. Muitas vezes, outros países esperam para ver como esse tipo de experimento funciona em os EUA Se funcionar bem, eles fazem a mesma coisa; se funcionar mal, eles tentam outra coisa.

2.6.3 O Loop Local: Modems, ADSL e Fibra

Agora é hora de iniciar nosso estudo detalhado de como funciona o sistema telefônico.

Vamos começar com a parte com a qual a maioria das pessoas está familiarizada: o local de dois fios loop vindo de uma estação final de companhia telefônica para as casas. O loop local é também conhecida como "última milha ", embora o comprimento possa ser de até várias milhas. Ele carregou informações analógicas por mais de 100 anos e é provável que

continuarão fazendo isso por alguns anos, devido ao alto custo de conversão para digital.

Muito esforço tem sido dedicado a espremer a rede de dados do cobre loops locais que já estão implantados. Os modems telefônicos enviam dados digitais para entre computadores no canal estreito que a rede telefônica fornece para um chamada de voz. Eles já foram amplamente usados, mas foram amplamente substituídos por tecnologias de banda larga como ADSL isso. reutilize o loop local para enviar digital dados de um cliente para a estação final, onde são desviados para a Internet.

Página 169

SEC. 2,6

A REDE PÚBLICA DE TELEFONE MUDADA

145

Ambos os modems e ADSL devem lidar com as limitações dos loops locais antigos: relargura de banda ativamente estreita, atenuação e distorção de sinais e suscetibilidade ao ruído elétrico, como diafonia.

Em alguns lugares, o loop local foi modernizado com a instalação de fibra óptica para (ou muito perto de) a casa. A fibra é o caminho do futuro. Essas instalações suportar redes de computadores desde o início, com o loop local tendo amplo largura de banda para serviços de dados. O fator limitante é o que as pessoas vão pagar, não o física do loop local.

Nesta seção, estudaremos o loop local, antigo e novo. Vamos cobrir modems de telefone, ADSL e fibra óptica para casa.

Modems de telefone

Para enviar bits pelo loop local, ou qualquer outro canal físico, eles devem ser convertidos em sinais analógicos que podem ser transmitidos pelo canal.

Esta conversão é realizada usando os métodos de modulação digital que nós estudado na seção anterior. Na outra extremidade do canal, o sinal analógico é convertido novamente em bits.

Um dispositivo que converte entre um fluxo de bits digitais e um sinal analógico que representa os bits é chamado um **modem**, que é a abreviação para "modulator demodulator." Os modems vêm em muitas variedades: modems de telefone, modems DSL, cabo modems, modems sem fio, etc. O modem pode ser integrado ao computador (que agora é comum para modems de telefone) ou ser uma caixa separada (que é comum para modems DSL e a cabo). Logicamente, o modem é inserido entre o computador (digital) e o sistema telefônico (análogo), conforme mostrado na Figura 2-32.

Fim
escritório
Codec
Modem
Computador
Loop local
(análogo)
Tronco (digital, fibra)
Linha digital
Linha analógica
Codec
Modem
ISP 1
ISP 2

Figura 2-32. O uso de transmissão analógica e digital para um computador chamada para o computador. A conversão é feita pelos modems e codecs.

Os modems de telefone são usados para enviar bits entre dois computadores através de um linha telefônica de grau de voz, no lugar da conversa que geralmente preenche a linha.

A principal dificuldade em fazer isso é que uma linha telefônica de voz é limitada a 3100 Hz, sobre o que é suficiente para manter uma conversa. Essa largura de banda é mais do que quatro ordens de magnitude a menos do que a largura de banda que é usada para Ethernet ou

Página 170

146

A CAMADA FÍSICA

INDIVÍDUO. 2

802.11 (WiFi). Sem surpresa, as taxas de dados dos modems de telefone também são quatro ordens de magnitude menores do que Ethernet e 802.11.

Vamos analisar os números para ver por que isso ocorre. O teorema de Nyquist diz nós que, mesmo com uma linha perfeita de 3000 Hz (o que uma linha telefônica decididamente não é),

não faz sentido enviar símbolos a uma taxa mais rápida do que 6000 baud. Na prática, a maioria dos modems envia a uma taxa de 2.400 símbolos / s, ou 2.400 baud, e se concentra em obter-

ajustar vários bits por símbolo, permitindo o tráfego em ambas as direções ao mesmo tempo (usando frequências diferentes para direções diferentes).

O humilde modem de 2400 bps usa 0 volts para um 0 lógico e 1 volt para um logi-cal 1, com 1 bit por símbolo. Um passo à frente, ele pode usar quatro símbolos diferentes, como em as quatro fases do QPSK, então com 2 bits / símbolo pode obter uma taxa de dados de 4800 bps.

Uma longa progressão de taxas mais altas foi alcançada à medida que a tecnologia se improuvado. Taxas mais altas requerem um conjunto maior de símbolos ou **constelações**. Com muitos símbolos, mesmo uma pequena quantidade de ruído na amplitude ou fase detectada pode resultar em um erro. Para reduzir a chance de erros, os padrões para alta velocidade modems usam alguns dos símbolos para correção de erros. Os esquemas são conhecidos como

TCM (Trellis Coded Modulation) (Ungerboeck, 1987).

O padrão de modem **V.32** usa 32 pontos de constelação para transmitir 4 bits de dados e 1 bit de verificação por símbolo a 2400 baud para atingir 9600 bps com erro de correção. A próxima etapa acima de 9600 bps é 14.400 bps. É chamado **V.32 bis** e transmite 6 bits de dados e 1 bit de verificação por símbolo a 2400 baud. Depois vem o **V.34**, que atinge 28.800 bps transmitindo 12 bits de dados / símbolo a 2400 baud. o constelação agora tem milhares de pontos. O modem final desta série é o **V.34 bis** que usa 14 bits de dados / símbolo a 2400 baud para atingir 33.600 bps.

Por que parar aqui? A razão pela qual os modems padrão param em 33.600 é que o O limite de Shannon para o sistema telefônico é de cerca de 35 kbps com base na média comprimento dos loops locais e a qualidade dessas linhas. Indo mais rápido do que isso seria violar as leis da física (departamento de termodinâmica).

No entanto, existe uma maneira de mudarmos a situação. No telefone escritório final da empresa, os dados são convertidos em formato digital para transmissão dentro a rede telefônica (o núcleo da rede telefônica convertido de analógico para digital há muito tempo). O limite de 35 kbps é para a situação em que há dois loops locais, um em cada extremidade. Cada um deles adiciona ruído ao sinal. Se nós pudéssemos se livrar de um desses loops locais, aumentaríamos o SNR e o máximo a taxa seria duplicada.

Essa abordagem é como os modems de 56 kbps são feitos para funcionar. Uma extremidade, normalmente um ISP, obtém uma alimentação digital de alta qualidade da estação final mais próxima. Assim, quando uma extremidade da conexão é um sinal de alta qualidade, como acontece com a maioria dos ISPs agora, o a taxa máxima de dados pode chegar a 70 kbps. Entre dois usuários domésticos com modems e linhas analógicas, o máximo ainda é 33,6 kbps. A razão pela qual os modems de 56 kbps (em vez de modems de 70 kbps) estão em uso tem a ver com o teorema de Nyquist. Um canal de telefone é transportado dentro do tele sistema de telefone como amostras digitais. Cada canal de telefone tem 4000 Hz de largura quando

Europa, todos os 8 bits estão disponíveis para os usuários, então os modems de 64.000 bits / s podem ter

usado, mas para obter um acordo internacional sobre um padrão, 56.000 foram escolhidos.

O resultado final são os padrões de modem **V.90** e **V.92**. Eles fornecem um Canal downstream de 56 kbps (ISP para o usuário) e um canal upstream de 33,6 kbps e 48 kbps canal (usuário para ISP), respectivamente. A assimetria é porque geralmente há mais dados transportados do ISP para o usuário do que de outra forma. Também significa que mais da largura de banda limitada pode ser alocada para o canal downstream para aumentar as chances de realmente funcionar a 56 kbps.

Digital Subscriber Lines

Quando a indústria de telefonia finalmente atingiu 56 kbps, deu um tapinha nas costas por um trabalho bem feito. Enquanto isso, a indústria de TV a cabo estava oferecendo velocidades de até

10 Mbps em cabos compartilhados. Como o acesso à Internet se tornou cada vez mais importante parte de seus negócios, as empresas de telefonia (LECs) começaram a perceber que precisavam um produto mais competitivo. A resposta deles foi oferecer novos serviços digitais sobre o loop local.

Inicialmente, havia muitas ofertas de alta velocidade sobrepostas, todas sob a geral nome de **xDSL** (**Digital Subscriber Line**), para vários *x* . Serviços com mais largura de banda do que o serviço de telefone padrão às vezes são chamados de **banda larga** , embora o termo realmente seja mais um conceito de marketing do que uma técnica específica conceito. Mais tarde, discutiremos o que se tornou o mais popular desses serviços vícios, **ADSL** (**DSL assimétrico**). Também usaremos o termo DSL ou xDSL como abreviação para todos os sabores.

A razão pela qual os modems são tão lentos é que os telefones foram inventados para carros rindo a voz humana e todo o sistema foi cuidadosamente otimizado para este objetivo. Data sempre foi enteado. No ponto onde cada loop local termina na estação final, o fio passa por um filtro que atenua todos os freios frequências abaixo de 300 Hz e acima de 3400 Hz. O corte não é nítido - 300 Hz e 3400 Hz são os pontos de 3 dB - portanto, a largura de banda é geralmente estimada em 4000 Hz, mesmo embora a distância entre os pontos de 3 dB seja 3100 Hz. Os dados na transmissão são, portanto, também restrito a esta faixa estreita.

O truque que faz o xDSL funcionar é que, quando um cliente assina, a linha de entrada é conectada a um tipo diferente de switch, que não tem este filtro, disponibilizando assim toda a capacidade do loop local. O limitante fator então se torna a física do loop local, que suporta cerca de 1 MHz, não a largura de banda artificial de 3100 Hz criada pelo filtro.

Infelizmente, a capacidade do loop local cai rapidamente com a distância da estação final conforme o sinal é cada vez mais degradado ao longo do fio. Isso também depende da espessura e da qualidade geral do par trançado. Um enredo do

Mbps

Figura 2-33. Largura de banda versus distância sobre UTP Categoria 3 para DSL.

A implicação dessa figura cria um problema para a companhia telefônica.

Quando ele escolhe uma velocidade para oferecer, ele está simultaneamente escolhendo um raio de sua extremidade

escritórios além dos quais o serviço não pode ser oferecido. Isso significa que quando distante clientes tentarem se inscrever para o serviço, eles podem ser informados de "Muito obrigado por seu interesse, mas você mora 100 metros longe da estação final mais próxima para obter este serviço. Poderia se mover?" Quanto menor for a velocidade escolhida, maior será o raio e mais clientes são cobertos. Mas quanto menor a velocidade, menos atraente o serviço é e menos pessoas estarão dispostas a pagar por ele.

É aqui que os negócios encontram a tecnologia.

Os serviços xDSL foram todos projetados com determinados objetivos em mente. Primeiro,

os serviços devem funcionar sobre os loops locais de par trançado Categoria 3 existentes. Sec-

em segundo lugar, eles não devem afetar os telefones e aparelhos de fax existentes dos

clientes. Terceiro,

eles devem ser muito mais rápidos do que 56 kbps. Quarto, eles devem estar sempre ligados, com apenas uma cobrança mensal e sem cobrança por minuto.

Para cumprir os objetivos técnicos, o espectro de 1,1 MHz disponível no lacete local

é dividido em 256 canais independentes de 4312,5 Hz cada. Este arranjo é

mostrado na Figura 2-34. O esquema OFDM, que vimos na seção anterior, é

usado para enviar dados através desses canais, embora seja freqüentemente chamado

de **DMT (Discrete**

MultiTone) no contexto de ADSL. Canal 0 é usado para **POTS (Plain Old**

Serviço telefônico). Os canais 1–5 não são usados, para manter o sinal de voz e dados

nals de interferir uns com os outros. Dos 250 canais restantes, um é usado

para controle upstream e um é usado para controle downstream. O resto está disponível

capaz de dados do usuário.

Em princípio, cada um dos canais restantes pode ser usado para dados full-duplex

stream, mas harmônicos, crosstalk e outros efeitos mantêm os sistemas práticos bem

Página 173

SEC. 2,6

A REDE PÚBLICA DE TELEFONE MUDADA

149

Po

W

e

r

Voz

Rio acima

Rio abaixo

256 canais de 4 kHz

0

25

1100 kHz

Figura 2-34. Operação de ADSL usando modulação multiton discrete.

abaixo do limite teórico. Cabe ao provedor determinar quantos canais

nels são usados para upstream e quantos para downstream. Uma mistura 50/50 de

upstream e downstream é tecnicamente possível, mas a maioria dos provedores alocam

algo como 80-90% da largura de banda para o canal downstream, já que a maioria

os usuários baixam mais dados do que carregam. Esta escolha dá origem ao "A" em

ADSL. Uma divisão comum é de 32 canais para upstream e o restante para downstream. isto

também é possível ter alguns dos canais upstream mais altos bidirecionais para

maior largura de banda, embora fazer essa otimização exija a adição de um

círculo para cancelar ecos.

O padrão internacional ADSL, conhecido como **G.dmt**, foi aprovado em 1999. Ele

permite velocidades de até 8 Mbps downstream e 1 Mbps upstream. isso foi

substituída por uma segunda geração em 2002, chamada ADSL2, com vários

provas para permitir velocidades de até 12 Mbps downstream e 1 Mbps up-

corrente. Agora temos ADSL2 +, que dobra a velocidade de downstream para 24

Mbps dobrando a largura de banda para usar 2,2 MHz sobre o par trançado. No entanto, os números citados aqui são as velocidades de melhor caso para boas linhas fechadas (dentro de 1 a 2 km) para a troca. Poucas linhas suportam essas taxas, e poucos provedores oferecem essas velocidades. Normalmente, os provedores oferecem algo como 1 Mbps downstream e 256 kbps upstream (serviço padrão), 4 Mbps downstream e 1 Mbps upstream (serviço aprimorado) e 8 Mbps downstream e 2 Mbps upstream (serviço premium).

Dentro de cada canal, a modulação QAM é usada a uma taxa de aproximadamente 4000 símbolos / s. A qualidade da linha em cada canal é constantemente monitorada e a taxa de dados é ajustada usando uma constelação maior ou menor, como aquelas em Fig. 2-23. Canais diferentes podem ter taxas de dados diferentes, com até 15 bits por símbolo enviado em um canal com um SNR alto e baixo para 2, 1 ou nenhum bit por símbolo bol enviado em um canal com um SNR baixo dependendo do padrão.

Um arranjo ADSL típico é mostrado na Figura 2-35. Neste esquema, um técnico da companhia telefônica deve instalar um **NID** (**Dispositivo de Interface de Rede**) no nas instalações do cliente. Esta pequena caixa de plástico marca o fim da comunicação telefônica propriedade da empresa e início da propriedade do cliente. Perto do NID (ou às vezes combinado com ele) é um **divisor** , um filtro analógico que separa o

Página 174

150

A CAMADA FÍSICA INDIVÍDUO. 2

Banda de 0–4000 Hz usada por POTS dos dados. O sinal POTS é encaminhado para o telefone ou fax existente. O sinal de dados é roteado para um modem ADSL, que usa processamento de sinal digital para implementar OFDM. Uma vez que a maioria ADSL os modems são externos, o computador deve estar conectado a eles em alta velocidade. Normalmente, isso é feito usando Ethernet, um cabo USB ou 802.11.

DSLAM
Divisor
Codec
Divisor
Telefone
Para ISP
ADSL
modem
Ethernet
Computador
Telefone
linha
Estação final da companhia telefônica
Instalações do cliente
Voz
interruptor
NID

Figura 2-35. Uma configuração típica de equipamento ADSL.

Na outra extremidade do fio, do lado da estação final, um divisor correspondente é instalado. Aqui, a porção de voz do sinal é filtrada e enviada para o chave de voz mal. O sinal acima de 26 kHz é encaminhado para um novo tipo de chamada de dispositivo

ed um **DSLAM** (**Digital Subscriber Line Access Multiplexer**), que contém o mesmo tipo de processador de sinal digital que o modem ADSL. Uma vez que os bits sido recuperado do sinal, os pacotes são formados e enviados para o ISP.

Esta separação completa entre o sistema de voz e ADSL torna relativamente fácil para uma companhia telefônica implantar o ADSL. Tudo o que é necessário é comprar conectar um DSLAM e divisor e conectar os assinantes ADSL ao divisor.

Outros serviços de alta largura de banda (por exemplo, ISDN) exigem mudanças muito maiores no equipamento de comutação existente.

Uma desvantagem do projeto da Fig. 2-35 é a necessidade de um NID e divisor nas instalações do cliente. A instalação só pode ser feita por tecnico da empresa, necessitando de uma "rolagem de caminhão" cara (ou seja, o envio de uma tecnico às instalações do cliente). Portanto, um design alternativo, sem divisor, chamado informalmente de **G.lite** , também foi padronizado. É o mesmo que a Fig. 2-35

mas sem o divisor do cliente. A linha telefônica existente é usada como está. A única diferença é que um microfiltro deve ser inserido em cada tomada de telefone

Página 175

SEC. 2,6

A REDE PÚBLICA DE TELEFONE MUDADA

151

entre o telefone ou modem ADSL e o fio. O microfiltro para o telefone é um filtro passa-baixa que elimina frequências acima de 3400 Hz; o microfiltro para o modem ADSL é um filtro passa-alta eliminando frequências abaixo de 26 kHz. No entanto, este sistema não é tão confiável quanto ter um divisor, então G.lite pode ser usado apenas até 1,5 Mbps (contra 8 Mbps para ADSL com um divisor). Para mais informações sobre ADSL, consulte Starr (2003).

Fibra para casa

Os loops locais de cobre implantados limitam o desempenho do ADSL e do telefone modems. Para deixá-los fornecer serviços de rede melhores e mais rápidos, empresas estão atualizando os loops locais em todas as oportunidades, instalando fibra óptica em todos

o caminho para casas e escritórios. O resultado é denominado **FttH (Fiber To The Home)**. Embora a tecnologia FttH esteja disponível há algum tempo, as implantações apenas começaram para decolar em 2005 com o crescimento da demanda por Internet de alta velocidade da usuários acostumados com DSL e cabo que queriam baixar filmes. Cerca de 4% de Casas nos EUA agora estão conectadas ao FttH com velocidades de acesso à Internet de até 100 Mbps.

Várias variações da forma " FttX " (onde X representa o porão, meio-fio, ou vizinhança) existem. Eles são usados para observar que a implantação da fibra pode chegar perto da casa. Neste caso, cobre (par trançado ou cabo coaxial) provê velocidades rápidas o suficiente na última curta distância. A escolha de quanto longe colocar a fibra é econômica, equilibrando o custo com a receita esperada. Em qualquer caso, a questão é que a fibra óptica cruzou a barreira tradicional da " última milha ".

Vamos nos concentrar em FttH em nossa discussão.

Como os fios de cobre anteriores, o loop local da fibra é passivo. Isso significa que não equipamento alimentado é necessário para amplificar ou de outra forma processar sinais. A fibra simplesmente transporta sinais entre a casa e a estação final. Isso, por sua vez, reduz custo e melhora a confiabilidade.

Normalmente, as fibras das casas são unidas de modo que apenas uma única fibra chega à estação final por grupo de até 100 casas. No direcionamento, divisores ópticos dividem o sinal da estação final para que ele alcance todos as casas. A criptografia é necessária para segurança se apenas uma casa deve ser capaz de decodificar o sinal. Na direção a montante, combinadores ópticos fundem os sinais das casas em um único sinal que é recebido na estação final.

Esta arquitetura é chamada de **PON (Passive Optical Network)**, e é mostrada na Figura 2-36. É comum usar um comprimento de onda compartilhado entre todas as casas para transmissão downstream e outro comprimento de onda para transmissão upstream. Mesmo com a divisão, a enorme largura de banda e a baixa atenuação de fibra significa que os PONs podem fornecer altas taxas para usuários em distâncias de até 20 km. As taxas de dados reais e outros detalhes dependem do tipo de PON. Dois tipos são comuns. **GPONs (PONs com capacidade de Gigabit)** vêm do mundo das telecomunicações comunicações, por isso são definidas por um padrão da ITU. **EPONs (Ethernet PONs)**

Página 176

152

A CAMADA FÍSICA

INDIVÍDUO. 2

Fibra

Ótico

divisor / combinador

Terminar escritório

Resto de
rede

Figura 2-36. Rede óptica passiva para Fiber To The Home.

estão mais sintonizados com o mundo das redes, por isso são definidos por um IEEE padrão. Ambos rodam em torno de um gigabit e podem transportar tráfego para serviços diferentes, incluindo Internet, vídeo e voz. Por exemplo, GPONs fornecem 2,4 Gbps downstream e 1,2 ou 2,4 Gbps upstream.

É necessário algum protocolo para compartilhar a capacidade da fibra única no final escritório entre as diferentes casas. A direção a jusante é fácil. O fim escritório pode enviar mensagens para cada casa diferente na ordem que desejar. No direção upstream, no entanto, as mensagens de casas diferentes não podem ser enviadas no ao mesmo tempo, ou sinais diferentes colidiriam. As casas também não podem ouvir cada outras transmissões para que não possam ouvir antes de transmitir. A solução é que o equipamento nas casas solicita e recebe slots de tempo para uso pelo equipamento na estação final. Para que isso funcione, há um processo de variação para ajustar os tempos de transmissão das casas para que todos os sinais recebidos no final escritório são sincronizados. O design é semelhante aos modems a cabo, que abordamos mais tarde neste capítulo. Para obter mais informações sobre o futuro dos PONs, consulte Grobe e Elbers (2008).

2.6.4 Troncos e Multiplexação

Os troncos na rede telefônica não são apenas muito mais rápidos do que os loops locais, eles são diferentes em dois outros aspectos. O núcleo da rede telefônica carrega informações digitais, não informações analógicas; isto é, bits, não voz. Este necessário indica uma conversão na estação final para a forma digital para transmissão ao longo dos troncos de transporte. Os troncos transportam milhares, até milhões, de chamadas simultaneamente. Esse compartilhamento é importante para a obtenção de economias de escala, uma vez que custa essencial

praticamente a mesma quantia de dinheiro para instalar e manter um tronco de alta largura de banda que

um tronco de baixa largura de banda entre dois escritórios de comutação. É realizado com versões de multiplexação TDM e FDM.

Abaixo, examinaremos brevemente como os sinais de voz são digitalizados para que pode ser transportado pela rede telefônica. Depois disso, veremos como o TDM é usado para transportar bits em troncos, incluindo o sistema TDM usado para fibra óptica

Página 177

SEC. 2,6

A REDE PÚBLICA DE TELEFONE MUDADA

153

(SONET). Em seguida, nos voltaremos para o FDM, uma vez que é aplicado à fibra óptica, que é chamada

multiplexação por divisão de comprimento de onda ed.

Digitalização de sinais de voz

No início do desenvolvimento da rede telefônica, o núcleo gerenciava a voz chamadas como informações analógicas. As técnicas de FDM foram usadas por muitos anos para multi

Canais de voz plex 4000 Hz (compostos de 3100 Hz mais bandas de guarda) em maiores e unidades maiores. Por exemplo, 12 chamadas na banda de 60 kHz a 108 kHz são conhecidas como um **grupo** e cinco grupos (um total de 60 chamadas) são conhecidos como um **supergrupo**, e em breve. Esses métodos FDM ainda são usados em alguns fios de cobre e microondas canais. No entanto, o FDM requer circuitos analógicos e não pode ser feito por um computador. Em contraste, o TDM pode ser manipulado inteiramente por dispositivos eletrônicos

tronics, por isso se tornou muito mais difundido nos últimos anos. Já que o TDM pode ser usado apenas para dados digitais e os loops locais produzem sinais analógicos, é necessário converter de analógico para digital na estação final, onde todos os indivíduos os loops locais se unem para serem combinados em troncos de saída.

Os sinais analógicos são digitalizados na estação final por um dispositivo chamado **codec**

(abreviação de "co DER dezembro oder"). O codec faz 8.000 amostras por segundo (125 μ sec / amostra) porque o teorema de Nyquist diz que isso é suficiente para capturar todas as informações da largura de banda do canal telefônico de 4 kHz. Em um sam-taxa de pling, informações seriam perdidas; em um nível superior, nenhuma informação extra iria ser ganho. Cada amostra da amplitude do sinal é quantizada em 8 bits número.

Esta técnica é chamada de **PCM** (**Pulse Code Modulation**). Forma o coração do sistema telefônico moderno. Como consequência, praticamente todos os intervalos de tempo dentro do sistema telefônico são múltiplos de 125 μ seg. O padrão a taxa de dados não compactados para uma chamada telefônica de nível de voz é, portanto, de 8 bits a cada 125 μ sec ou 64 kbps.

Na outra extremidade da chamada, um sinal analógico é recriado a partir do quantizado samples tocando-os (e suavizando-os) ao longo do tempo. Não será ex-na verdade o mesmo que o sinal analógico original, embora tenhamos amostrado no Taxa de Nyquist, porque as amostras foram quantizadas. Para reduzir o erro devido a quantização, os níveis de quantização são espaçados desigualmente. Uma escala logarítmica é usado que fornece relativamente mais bits para amplitudes de sinal menores e menos bits para grandes amplitudes de sinal. Desta forma, o erro é proporcional ao amplitude do sinal.

Duas versões de quantização são amplamente utilizadas: **μ -law**, usado na América do Norte e Japão, e **A-law**, usado na Europa e no resto do mundo. Ambas as versões são especificado na norma ITU G.711. Uma maneira equivalente de pensar sobre esse processo é imaginar que a faixa dinâmica do sinal (ou a relação entre as maiores e os menores valores possíveis) é compactado antes de ser (uniformemente) quantizado, e em seguida, expandido quando o sinal analógico é recriado. Por isso é chamado

Página 178

154

A CAMADA FÍSICA

INDIVÍDUO. 2

companding. Também é possível comprimir as amostras depois de digitalizadas de modo que exigem muito menos do que 64 kbps. No entanto, vamos deixar este tópico para quando exploramos aplicativos de áudio, como voz sobre IP.

Multiplexação por divisão de tempo

TDM baseado em PCM é usado para transportar várias chamadas de voz sobre troncos por envio-ing uma amostra de cada chamada a cada 125 μ seg. Quando a transmissão digital começou emergindo como uma tecnologia viável, a ITU (então chamada de CCITT) foi incapaz de alcançar acordo sobre um padrão internacional para PCM. Consequentemente, uma variedade de esquemas incompatíveis estão agora em uso em diferentes países ao redor do mundo.

O método usado na América do Norte e no Japão é a portadora **T1**, representada em Fig. 2-37. (Tecnicamente falando, o formato é denominado DS1 e a operadora é chamada ed T1, mas seguindo a tradição da indústria generalizada, não vamos tornar isso sutil distinção aqui.) A portadora T1 consiste em 24 canais de voz multiplexados em conjunto. dela. Cada um dos 24 canais, por sua vez, consegue inserir 8 bits no fluxo de saída.

Canal

1

Canal

2

Canal

3

Canal

4

Canal

24

Quadro de 193 bits (125 μ seg)

7 dados

bits por

canal

por amostra

Bit 1 é

um enquadramento

código

O bit 8 é para
sinalização
0
1

Figura 2-37. A portadora T1 (1,544 Mbps).

Um quadro consiste em $24 \times 8 = 192$ bits mais um bit extra para fins de controle, rendendo 193 bits a cada 125 μ seg. Isso dá uma taxa de dados bruta de 1,544 Mbps, de qual 8 kbps é para sinalização. O 193º bit é usado para sincronização de quadros e sinalização. Em uma variação, o 193º bit é usado em um grupo de 24 quadros de chamada ed um **superquadro estendido**. Seis dos bits, na 4ª, 8ª, 12ª, 16ª, 20ª e 24ª posições, assuma o padrão alternado 001011. . . . Normalmente, o receptor continua verificando esse padrão para ter certeza de que ele não perdeu a sincronização. Mais seis bits são usados para enviar um código de verificação de erro para ajudar o receptor a confirmar que está sincronizado. Se sair de sincronia, o receptor pode procurar o padrão tern e validar o código de verificação de erro para obter a ressincronização. Os 12 restantes

Página 179

SEC. 2,6

A REDE PÚBLICA DE TELEFONE MUDADA

155

bits são usados para informações de controle para operação e manutenção da rede, como relatórios de desempenho da extremidade remota. O formato T1 tem várias variações. As versões anteriores enviaram sinalização informações **dentro da banda**, ou seja, no mesmo canal que os dados, usando alguns dos os bits de dados. Este projeto é uma forma de **sinalização associada ao canal**, porque cada canal tem seu próprio subcanal de sinalização privado. Em um arranjo, o o bit menos significativo de uma amostra de 8 bits em cada canal é usado a cada seis quadro, Armação. Ele tem o nome colorido de **sinalização de bits roubados**. A ideia é que alguns bits roubados não importam para chamadas de voz. Ninguém vai ouvir a diferença. Para os dados, no entanto, é outra história. Entregar os bits errados não ajuda, para dizer o mínimo. Se versões mais antigas do T1 são usadas para transportar dados, apenas 7 de 8 bits, ou 56 kbps podem ser usados em cada um dos 24 canais. Em vez disso, as versões mais recentes do T1 fornecer canais claros nos quais todos os bits possam ser usados para enviar dados. Claro canais são o que as empresas que alugam uma linha T1 desejam quando enviam dados através a rede telefônica no lugar de amostras de voz. A sinalização para qualquer chamada de voz é em seguida, tratado **fora da banda**, ou seja, em um canal separado dos dados. Frequentemente, a sinalização é feita com **sinalização de canal comum** em que há uma canal de sinalização. Um dos 24 canais pode ser usado para este propósito. Fora da América do Norte e do Japão, a portadora **E1** de 2.048 Mbps é usada de T1. Esta portadora tem 32 amostras de dados de 8 bits compactadas em 125 - μ sec quadro, Armação. Trinta dos canais são usados para informações e até dois são usados para sinalização. Cada grupo de quatro quadros fornece 64 bits de sinalização, metade dos quais são usado para sinalização (seja associada ao canal ou canal comum) e metade de que são usados para sincronização de quadros ou são reservados para cada país usar como deseja. A multiplexação por divisão de tempo permite que várias portadoras T1 sejam multiplexadas em operadoras de ordem superior. A Figura 2-38 mostra como isso pode ser feito. À esquerda vemos quatro canais T1 sendo multiplexados em um canal T2. A multiplexação em T2 e acima é feito bit a bit, em vez de byte a byte com os 24 canais de voz que compõem um quadro T1. Quatro fluxos T1 a 1,544 Mbps devem gerar 6,176 Mbps, mas T2 é na verdade 6,312 Mbps. Os bits extras são usados para enquadrar e re-covery no caso de o portador escorregar. T1 e T3 são amplamente utilizados pelos clientes, enquanto T2 e T4 são usados apenas dentro do próprio sistema de telefonia, então eles não estão bem conhecido. No próximo nível, sete fluxos T2 são combinados bit a bit para formar um fluxo T3.

Em seguida, seis fluxos T3 são unidos para formar um fluxo T4. Em cada etapa, uma pequena quantidade

de sobrecarga é adicionado para enquadramento e recuperação no caso de a sincronização ser entre o remetente e o destinatário é perdido.

Assim como há pouco acordo sobre a transportadora básica entre os Estados Unidos e no resto do mundo, há igualmente pouco acordo sobre como é ser multiplexado em portadoras de maior largura de banda. O esquema dos EUA de intensificação em 4, 7, e 6 não pareceu a todos como o caminho a seguir, então o padrão da ITU exige multiplexar quatro fluxos em um fluxo em cada nível. Além disso, o enquadramento e

Página 180

156

A CAMADA FÍSICA INDIVÍDUO. 2

6 5 4 3 2 1 0
5 1
4 0
6 2
7 3
6: 1
7: 1
4: 1
4 fluxos T1 em
1 T2 stream out
6,312 Mbps
T2
1,544 Mbps
T1
44,736 Mbps
T3
274,176 Mbps
T4
7 fluxos T2 em
6 fluxos T3 em

Figura 2-38. Multiplexação de fluxos T1 em portadoras superiores.

os dados de recuperação são diferentes nos padrões dos EUA e da ITU. A hierarquia ITU para 32, 128, 512, 2048 e 8192 canais funcionam a velocidades de 2.048, 8.848, 34.304, 139.264 e 565.148 Mbps.

SONET / SDH

Nos primeiros dias da fibra óptica, cada companhia telefônica tinha sua própria sistema TDM óptico proprietário. Depois que a AT&T foi desmembrada em 1984, a televisão local as empresas de telefonia tiveram que se conectar a várias operadoras de longa distância, todas com diferenças

diferentes sistemas ópticos TDM, então a necessidade de padronização tornou-se óbvia. No 1985, Bellcore, o braço de pesquisa do RBOC, começou a trabalhar em um padrão, chamado **SONET (Synchronous Optical NETwork)**.

Mais tarde, a ITU juntou-se ao esforço, o que resultou em um padrão SONET e um conjunto de recomendações paralelas do ITU (G.707, G.708 e G.709) em 1989. O ITU as recomendações são chamadas de **SDH (Synchronous Digital Hierarchy)**, mas diferem da SONET apenas em pequenas maneiras. Praticamente todo o tráfego telefônico de longa distância nos Estados Unidos, e muitos deles em outros lugares, agora usam troncos executando SONET na camada física. Para obter informações adicionais sobre SONET, consulte Bellamy (2000), Goralski (2002) e Shepard (2001).

O projeto SONET tinha quatro objetivos principais. Em primeiro lugar, a SONET teve que possibilitar a interação entre diferentes transportadoras. É necessário alcançar este objetivo definir um padrão de sinalização comum com relação ao comprimento de onda, tempo, quadro estrutura operacional e outras questões.

Em segundo lugar, alguns meios foram necessários para unificar os EUA, Europa e Japão sistemas digitais, todos baseados em canais PCM de 64 kbps, mas combinados de maneiras diferentes (e incompatíveis).

Terceiro, a SONET precisava fornecer uma maneira de multiplexar vários canais digitais. Na época em que o SONET foi desenvolvido, a operadora digital de maior velocidade realmente usada

amplamente nos Estados Unidos foi T3, a 44,736 Mbps. T4 foi definido, mas não usado

SEC. 2,6

A REDE PÚBLICA DE TELEFONE MUDADA

157

muito, e nada foi definido mesmo acima da velocidade T4. Parte da missão da SONET era continuar a hierarquia para gigabits / s e além. Uma maneira padrão de canais mais lentos tiplex em um canal SONET também eram necessários.

Quarto, a SONET teve que fornecer suporte para operações, administração e manutenção (OAM), que são necessários para gerenciar a rede. Sistemas anteriores não fez isso muito bem.

Uma decisão inicial foi tornar o SONET um sistema TDM tradicional, com o largura de banda inteira da fibra dedicada a um canal contendo slots de tempo para o vários subcanais. Como tal, o SONET é um sistema síncrono. Cada remetente e receptor está vinculado a um relógio comum. O relógio mestre que controla o sistema tem uma precisão de cerca de 1 parte em 10^9 . Bits em uma linha SONET são enviados no extremo intervalos precisos, controlados pelo relógio mestre.

O quadro SONET básico é um bloco de 810 bytes enviado a cada 125 μ s.

Uma vez que o SONET é síncrono, os quadros são emitidos independentemente de haver ou não dados úteis para enviar. Ter 8.000 quadros / s corresponde exatamente à taxa de amostragem de os canais PCM usados em todos os sistemas de telefonia digital.

Os quadros SONET de 810 bytes são melhor descritos como um retângulo de bytes, 90 colunas de largura por 9 linhas de altura. Assim, $8 \times 810 = 6480$ bits são transmitidos 8000 vezes por segundo, para uma taxa de dados bruta de 51,84 Mbps. Este layout é o básico Canal SONET, denominado **STS-1 (Synchronous Transport Signal-1)**. Todos SONET os troncos são múltiplos de STS-1.

As primeiras três colunas de cada quadro são reservadas para gerenciamento do sistema informações, conforme ilustrado na Figura 2-39. Neste bloco, as três primeiras linhas contêm a sobrecarga da seção; os próximos seis contêm a linha aérea. A seção aérea é gerado e verificado no início e no final de cada seção, enquanto a linha

A sobrecarga é gerada e verificada no início e no final de cada linha.

Um transmissor SONET envia quadros back-to-back de 810 bytes, sem lacunas entre entre eles, mesmo quando não há dados (nesse caso, ele envia dados fictícios).

Do ponto de vista do receptor, tudo o que ele vê é um fluxo contínuo de bits, então como ele sabe onde cada quadro começa? A resposta é que os primeiros 2 bytes de cada quadro contém um padrão fixo que o receptor procura. Se encontrar isso padrão no mesmo lugar em um grande número de quadros consecutivos, ele assume que está sincronizado com o remetente. Em teoria, um usuário poderia inserir este padrão no carga útil de forma regular, mas na prática isso não pode ser feito devido à multiplexação de vários usuários no mesmo quadro e por outros motivos.

As 87 colunas restantes de cada quadro contêm $87 \times 9 \times 8 \times 8000 = 50,112$ Mbps de dados do usuário. Esses dados do usuário podem ser amostras de voz, T1 e outras operadoras

engolido inteiro, ou pacotes. SONET é simplesmente um contêiner conveniente para transbits esportivos. O **SPE (Synchronous Payload Envelope)**, que transporta o usuário os dados nem sempre começam na linha 1, coluna 4. O SPE pode começar em qualquer lugar dentro do quadro. Um ponteiro para o primeiro byte está contido na primeira linha da linha a sobrecarga. A primeira coluna do SPE é a sobrecarga do caminho (ou seja, o cabeçalho para protocolo de subcamada de caminho ponta a ponta).

158

A CAMADA FÍSICA

INDIVÍDUO. 2

Sonet
quadro, Armação
(125 μ seg)

Sonet
quadro, Armação
(125 μ seg)

9
Linhas
...
...
87 colunas
3 colunas
para sobrecarga
SPE
Seção
a sobrecarga
Linha
a sobrecarga
Caminho
a sobrecarga

Figura 2-39. Dois quadros SONET consecutivos.

A capacidade de permitir que o SPE comece em qualquer lugar dentro do quadro SONET e mesmo para abranger dois quadros, como mostrado na Figura 2-39, dá mais flexibilidade ao sistema tem. Por exemplo, se uma carga chegar à fonte enquanto um SONET fictício quadro está sendo construído, ele pode ser inserido no quadro atual em vez de sendo retido até o início do próximo.

A hierarquia de multiplexação SONET / SDH é mostrada na Figura 2-40. Taxas de STS-1 a STS-768 foram definidos, variando de aproximadamente uma linha T3 a 40 Gbps. Taxas ainda mais altas certamente serão definidas ao longo do tempo, com OC-3072 a 160 Gbps sendo o próximo na linha se e quando for tecnologicamente viável. O opti-portadora cal correspondente a STS- n é chamada OC- n , mas é bit a bit o mesmo, exceto para um certo reordenamento de bits necessário para a sincronização. Os nomes SDH são diferentes diferente, e eles começam em OC-3 porque os sistemas baseados em ITU não têm uma taxa próxima 51,84 Mbps. Mostramos as taxas comuns, que procedem de OC-3 em múltiplos de quatro. A taxa de dados bruta inclui toda a sobrecarga. Os dados SPE a taxa exclui a sobrecarga de linha e seção. A taxa de dados do usuário exclui todos os cabeça e conta apenas as 87 colunas de carga útil.

Como um aparte, quando uma portadora, como OC-3, não é multiplexada, mas carrega o dados de apenas uma única fonte, a letra c (para concatenado) é anexada ao designação, então OC-3 indica uma portadora de 155,52 Mbps que consiste em três Portadoras OC-1, mas OC-3c indica um fluxo de dados de uma única fonte em 155,52 Mbps. Os três fluxos OC-1 dentro de um fluxo OC-3c são intercalados por coluna - primeira coluna 1 do fluxo 1, depois coluna 1 do fluxo 2 e, em seguida, coluna 1 do fluxo 3, seguido pela coluna 2 do fluxo 1 e assim por diante - levando a um quadro 270 colunas de largura e 9 linhas de profundidade.

Página 183

SEC. 2,6

A REDE PÚBLICA DE TELEFONE MUDADA

159

SONET

SDH

Taxa de dados (Mbps)

Elétrico

Ótico

Ótico

Bruto

SPE

Do utilizador

STS-1

OC-1

51,84

50,112

49.536

STS-3

OC-3

STM-1

155,52

150,336

148,608

STS-12

OC-12

STM-4
622,08
601.344
594.432
STS-48
OC-48
STM-16
2488,32
2405.376
2377,728
STS-192
OC-192
STM-64
9953,28
9621,504
9510.912
STS-768
OC-768
STM-256
39813,12
38486.016
38043.648

Figura 2-40. Taxas de multiplex SONET e SDH.

Wavelength Division Multiplexing

Uma forma de multiplexação por divisão de frequência é usada, bem como TDM para aproveitar a tremenda largura de banda dos canais de fibra óptica. É chamado **WDM (Wavelength division multiplexação por divisão de comprimento)**. O princípio básico do WDM nas fibras é descrito na Figura 2-41. Aqui, quatro fibras se unem em um combinador óptico, cada com sua energia presente em um comprimento de onda diferente. Os quatro feixes são combinados em uma única fibra compartilhada para transmissão a um destino distante. No final, o feixe é dividido em tantas fibras quantas havia no lado de entrada. Cada fibra contém um núcleo curto e especialmente construído que filtra todos, exceto um Comprimento de onda. Os sinais resultantes podem ser encaminhados para seu destino ou ed de maneiras diferentes para transporte multiplexado adicional.

Não há realmente nada de novo aqui. Esta forma de operação é apenas dividir a multiplexação de visão em frequências muito altas, com o termo WDM devido ao descrição dos canais de fibra óptica por seu comprimento de onda ou " cor " ao invés de frequência. Contanto que cada canal tenha sua própria faixa de frequência (ou seja, comprimento de onda)

e todos os intervalos são disjuntos, eles podem ser multiplexados juntos no longo curso de fibra. A única diferença com FDM elétrico é que um sistema óptico usando um grade de difração é completamente passiva e, portanto, altamente confiável.

A razão pela qual o WDM é popular é que a energia em um único canal é normalmente apenas alguns gigahertz de largura, porque esse é o limite atual de quanto rápido podemos converter entre sinais elétricos e ópticos. Executando muitos canais em paralelo em diferentes comprimentos de onda, a largura de banda agregada é aumentada linearmente com o número de canais. Uma vez que a largura de banda de uma única banda de fibra é de cerca de 25.000 GHz (ver Figura 2-7), teoricamente há espaço para 2.500 canais de 10 Gbps, mesmo em 1 bit / Hz (e taxas mais altas também são possíveis).

A tecnologia WDM tem progredido a um ritmo que coloca a tecnologia de computador gasto vergonha. WDM foi inventado por volta de 1990. Os primeiros sistemas comerciais tinha oito canais de 2,5 Gbps por canal. Em 1998, sistemas com 40 canais

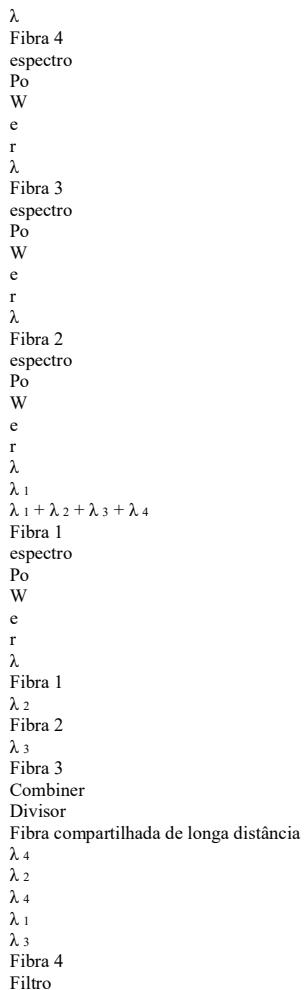


Figura 2-41. Wavelength Division Multiplexing.

de 2,5 Gbps estavam no mercado. Em 2006, havia produtos com 192 canais de 10 Gbps e 64 canais de 40 Gbps, capaz de mover até 2,56 Tbps. Isto a largura de banda é suficiente para transmitir 80 filmes em DVD completos por segundo. os canais também são compactados firmemente na fibra, com 200, 100 ou tão pouco quanto 50 GHz de separação. Demonstrações de tecnologia por empresas após o direito de se gabar mostraram 10 vezes essa capacidade no laboratório, mas indo do laboratório para o campo geralmente leva pelo menos alguns anos. Quando o número de canais é muito grande e os comprimentos de onda são espaçados próximos, o sistema é referido como **DWDM** (**Denso WDM**).

Um dos motores da tecnologia WDM é o desenvolvimento de componentes ópticos totalmente componentes. Antes, a cada 100 km era necessário dividir todos os canais e converter cada um em um sinal elétrico para amplificação separadamente antes de reconvertendo-os em sinais ópticos e combinando-os. Hoje em dia, totalmente óptico amplificadores podem regenerar todo o sinal uma vez a cada 1000 km sem a necessidade para múltiplas conversões optoeletrônicas.

No exemplo da Figura 2.41, temos um sistema de comprimento de onda fixo. Bits de a fibra de entrada 1 vai para a fibra de saída 3, os bits da fibra de entrada 2 vão para a fibra de saída 1, etc.

No entanto, também é possível construir sistemas WDM que são comutados no óptico domínio cal. Em tal dispositivo, os filtros de saída são ajustáveis usando Fabry-Perot ou Interferômetros Mach-Zehnder. Esses dispositivos permitem que as frequências selecionadas ser alterado dinamicamente por um computador de controle. Essa capacidade fornece uma grande quantidade de flexibilidade para provisionar muitos caminhos de comprimento de onda diferentes através da rede telefônica de um conjunto fixo de fibras. Para obter mais informações sobre óptica redes e WDM, consulte Ramaswami et al. (2009).

SEC. 2,6

A REDE PÚBLICA DE TELEFONE MUDADA

161

2.6.5 Troca

Do ponto de vista do engenheiro de telefonia médio, o sistema telefônico é dividido em duas partes principais: planta externa (os loops e troncos locais, uma vez que eles estão fisicamente fora das centrais de comutação) e dentro da planta (as centrais, que estão dentro das centrais de comutação). Acabamos de olhar para a planta externa. Agora é hora de examinar a planta interna.

Duas técnicas de comutação diferentes são usadas pela rede hoje em dia: circuito comutação e comutação de pacotes. O sistema telefônico tradicional é baseado em circuito comutação cuit, mas a comutação de pacotes está começando a fazer incursões com o surgimento de tecnologia de voz sobre IP. Iremos entrar em comutação de circuito com alguns detalhes e compare-o com a comutação de pacotes. Ambos os tipos de troca são importantes o suficiente que voltaremos a eles quando chegarmos à camada de rede.

Comutação de circuitos

Conceitualmente, quando você ou seu computador fazem uma chamada telefônica, o switch-equipamento dentro do sistema telefônico busca um caminho físico por todo o caminho do seu telefone para o telefone do receptor. Esta técnica é chamada de **circuito comutação**. Ele é mostrado esquematicamente na Figura 2.42 (a). Cada um dos seis retângulos representa uma estação de comutação de operadora (estação final, estação tarifada etc.). Neste exemplo, cada escritório tem três linhas de entrada e três linhas de saída. Quando uma chamada passa através de uma central de comutação, uma conexão física é (conceitualmente) estabelecida entre a linha em que a chamada entrou e uma das linhas de saída, conforme mostrado por as linhas pontilhadas.

Nos primeiros dias do telefone, a conexão era feita pela operadora conectar um cabo jumper nas tomadas de entrada e saída. Na verdade, uma surpreendente história está associada à invenção do equipamento de comutação automática de circuitos. Foi inventado por um agente funerário do século 19 no Missouri chamado Almon B. Strowger. Pouco depois de o telefone ser inventado, quando alguém morreu, um dos sobreviventes ligaria para a operadora da cidade e diria "Por favor, conecte-me a um agente funerário". Felizmente para o Sr. Strowger, havia dois agentes funerários em sua cidade, e a esposa de outro era a operadora de telefonia da cidade. Ele rapidamente viu que ele teria que inventar um equipamento de comutação telefônica automática ou indo à falência. Ele escolheu a primeira opção. Por quase 100 anos, o equipamento de comutação de circuitos usado em todo o mundo era conhecido como **engrenagem de Strowger**.

(Seu-tory não registra se a operadora de mesa agora desempregada conseguiu um emprego como operador de informações, respondendo a perguntas como "Qual é o número do telefone de um agente funerário?")

O modelo mostrado na Figura 2-42 (a) é altamente simplificado, é claro, porque as peças do caminho físico entre os dois telefones pode, na verdade, ser microondas ou links de fibra nos quais milhares de chamadas são multiplexadas. No entanto, o básico ideia é válida: uma vez que uma chamada foi estabelecida, um caminho dedicado entre ambas as extremidades

existe e continuará a existir até que a chamada seja concluída.

162

A CAMADA FÍSICA

INDIVÍDUO. 2

(uma)

(b)

Troca de escritório
Físico (cobre)

conexão configurada
quando a chamada é feita
Pacotes enfileirados
para subsequente
transmissão
Computador
Computador

Figura 2-42. (a) Comutação de circuitos. (b) Comutação de pacotes.

Uma propriedade importante da comutação de circuito é a necessidade de configurar um ponto a ponto caminho *antes que* qualquer dado possa ser enviado. O tempo decorrido entre o final da discagem e o início do toque pode ser facilmente de 10 segundos, mais em longa distância ou internacional chamadas. Durante este intervalo de tempo, o sistema telefônico está procurando um caminho, como mostrado na Figura 2-43 (a). Observe que antes mesmo de a transmissão de dados começar, a chamada

o sinal de solicitação deve se propagar até o destino e ser confirmado.

Para muitos aplicativos de computador (por exemplo, verificação de crédito de ponto de venda), configuração longa tempos são indesejáveis.

Como consequência do caminho reservado entre os chamadores, uma vez que o configuração foi concluída, o único atraso para os dados é o tempo de propagação para o sinal eletromagnético, cerca de 5 ms por 1000 km. Também como consequência do caminho estabelecido, não há perigo de congestionamento, ou seja, uma vez que a chamada foi colocada, você nunca recebe sinais de ocupado. Claro, você pode obter um antes do a conexão foi estabelecida devido à falta de capacidade de comutação ou tronco.

Comutação de pacotes

A alternativa para a comutação de circuitos é a **comutação de pacotes**, mostrada na Figura 2-42 (b) e descrito no Cap. 1. Com esta tecnologia, os pacotes são enviados assim que Eles estão disponíveis. Não há necessidade de configurar um caminho dedicado com antecedência, ao contrário de

SEC. 2,6

A REDE PÚBLICA DE TELEFONE MUDADA

163

Sinal de solicitação de chamada

Dados

AB

tronco

UMA

B

C

(uma)

D

UMA

B

C

(b)

D

AC

tronco

CD

tronco

Ligar

aceitar

sinal

Propagação

demora

Fila

demora

Pkt 1

Pkt 2

Pkt 3

Pkt 1

Pkt 2

Pkt 3

Pkt 1

Pkt 2

Pkt 3

Tempo

gasto

Caçando

para um
extrovertido
tronco
Tempo

Figura 2-43. Tempo de eventos em (a) comutação de circuitos, (b) comutação de pacotes.

com comutação de circuito. Cabe aos roteadores usar a transmissão store-and-forward para enviar cada pacote em seu caminho para o destino por conta própria. Este procedimento é ao contrário da comutação de circuito, em que o resultado da configuração da conexão é a reservação da largura de banda desde o emissor até o receptor. Todos os dados no circuito segue esse caminho. Entre outras propriedades, fazer com que todos os dados sigam o mesmo caminho significa que ele não pode chegar fora de ordem. Com a comutação de pacotes, existe

nenhum caminho fixo, portanto, pacotes diferentes podem seguir caminhos diferentes, dependendo da rede

condições de trabalho no momento em que são enviados e podem chegar fora de serviço.

As redes de comutação de pacotes colocam um limite superior estrito no tamanho dos pacotes.

Isso garante que nenhum usuário possa monopolizar qualquer linha de transmissão por muito tempo (por exemplo,

muitos milissegundos), para que as redes comutadas por pacotes possam lidar com o tráfego interativo

físico. Também reduz o atraso, pois o primeiro pacote de uma mensagem longa pode ser protegido antes que o segundo chegue totalmente. No entanto, o armazenamento e encaminhamento atraso de acumulação de um pacote na memória do roteador antes de ser enviado para o

Página 188

164

A CAMADA FÍSICA INDIVÍDUO. 2

próximo roteador excede o de comutação de circuito. Com a comutação de circuitos, os bits apenas fluem através do fio continuamente.

A comutação de pacotes e circuitos também difere de outras maneiras. Porque sem largura de banda é reservado com comutação de pacotes, os pacotes podem ter que esperar para serem encaminhados. Isso introduz **atraso na fila** e congestionamento se muitos pacotes forem enviados no mesmo tempo. Por outro lado, não há perigo de obter um sinal de ocupado e sendo incapaz de usar a rede. Assim, o congestionamento ocorre em momentos diferentes com comutação de circuitos (no momento da configuração) e comutação de pacotes (quando os pacotes são enviados).

Se um circuito foi reservado para um determinado usuário e não há tráfego, é largura de banda é desperdiçada. Não pode ser usado para outro tráfego. A comutação de pacotes faz não desperdiçar largura de banda e, portanto, é mais eficiente do ponto de vista do sistema. Sobreviver a essa compensação é crucial para compreender a diferença entre o circuito comutação e comutação de pacotes. A compensação é entre serviço garantido e desperdiçar recursos versus não garantir serviço e não desperdiçar recursos.

A comutação de pacotes é mais tolerante a falhas do que a comutação de circuitos. Na verdade, isso é

por que foi inventado. Se um interruptor cair, todos os circuitos que o usam são terminados e nenhum mais tráfego pode ser enviado em qualquer um deles. Com a comutação de pacotes, os pacotes podem ser roteados em torno de switches mortos.

Uma diferença final entre comutação de circuito e pacote é o algoritmo de cobrança ritmo. Com a comutação de circuitos, o carregamento tem sido historicamente baseado na distância e tempo. Para telefones celulares, a distância geralmente não desempenha um papel, exceto para chamadas internacionais, e o tempo desempenha apenas uma função grosseira (por exemplo, um plano de chamadas com 2.000

os minutos grátis custam mais do que um com 1000 minutos grátis e às vezes noites ou fins de semana são baratos). Com a comutação de pacotes, o tempo de conexão não é um problema, mas o

volume de tráfego é. Para usuários domésticos, os ISPs geralmente cobram uma taxa mensal fixa antes de

porque é menos trabalhoso para eles e seus clientes podem entender este modelo, mas as operadoras de backbone cobram as redes regionais com base no volume de seu tráfego. As diferenças estão resumidas na Figura 2-44. Tradicionalmente, a rede telefônica obras usaram comutação de circuito para fornecer chamadas telefônicas de alta qualidade, e as redes de computadores têm usado comutação de pacotes para simplicidade e eficiência. No entanto, existem exceções notáveis. Algumas redes de computadores mais antigas foram circuito comutado sob as tampas (por exemplo, X.25) e algumas redes telefônicas mais recentes usar comutação de pacotes com tecnologia de voz sobre IP. Isso se parece com um standard telefone externo para usuários, mas dentro da rede pacotes de voz os dados são trocados. Essa abordagem permitiu que as empresas iniciantes comercializassem chamadas internacionais baratas por meio de cartões telefônicos, embora talvez com qualidade de chamada inferior do que os operadores existentes.

2.7 O SISTEMA DE TELEFONE MÓVEL

O sistema telefônico tradicional, mesmo que algum dia receba multigigabit ponta a ponta fibra final, ainda não será capaz de satisfazer um grupo crescente de usuários: pessoas no vai. As pessoas agora esperam fazer ligações e usar seus telefones para verificar

Página 189

SEC. 2,7

O SISTEMA DE TELEFONE MÓVEL

165

Item

Círculo comutado

Pacote comutado

Configuração de chamada

Requeridos

Não é necessário

Caminho físico dedicado

sim

Não

Cada pacote segue a mesma rota

sim

Não

Os pacotes chegam em ordem

sim

Não

Um switch crash é fatal

sim

Não

Largura de banda disponível

Fixo

Dinâmico

Tempo de possível congestionamento

Na hora da configuração

Em cada pacote

Largura de banda potencialmente desperdiçada

sim

Não

Transmissão armazenar e encaminhar

Não

sim

Carregando

Por minuto

Por pacote

Figura 2-44. Uma comparação de redes comutadas por circuitos e redes comutadas por pacotes.

enviar e-mail e navegar na Web em aviões, carros, piscinas e enquanto faz jogging no Parque. Conseqüentemente, há um grande interesse na tecnologia sem fio telefonia. Nas seções a seguir, estudaremos esse tópico com alguns detalhes.

O sistema de telefonia móvel é usado para comunicação de voz e dados de área ampla ção. **Os telefones celulares** (às vezes chamados de **telefones celulares**) passaram por três gerações distintas, amplamente chamadas de **1G**, **2G** e **3G**. As gerações são:

1. Voz analógica.

2. Voz digital.

3. Voz e dados digitais (Internet, e-mail, etc.).

(Os telefones celulares não devem ser confundidos com **telefones sem fio** que consistem em um estação base e um aparelho vendido como um conjunto para uso doméstico. Estes nunca são usados para rede, portanto, não os examinaremos mais a fundo.)

Embora a maior parte de nossa discussão seja sobre a tecnologia desses sistemas é interessante notar como as decisões políticas e pequenas de marketing podem ter um grande impacto. O primeiro sistema móvel foi desenvolvido nos EUA pela AT&T e mandatada para todo o país pela FCC. Como resultado, todos os EUA tiveram um sistema único (análogo) e um telefone celular adquirido na Califórnia também funcionou em Nova York. Em contraste, quando os telefones celulares chegaram à Europa, cada país decretou visou seu próprio sistema, o que resultou em um fiasco.

A Europa aprendeu com seu erro e, quando surgiu o digital, o governo PTTs executados por ment se reuniram e padronizaram em um único sistema (GSM), para que qualquer

O telefone móvel europeu funcionará em qualquer lugar da Europa. Até então, os EUA tinham decaído

decidiu que o governo não deveria estar no negócio de padronização, então deixou o digital para o mercado. Esta decisão resultou em diferentes fabricantes de equipamentos produzindo diferentes tipos de telefones celulares. Como consequência, nos EUA

Página 190

166

A CAMADA FÍSICA

INDIVÍDUO. 2

dois principais - e completamente incompatíveis - sistemas de telefonia móvel digital eram implantado, bem como outros sistemas menores.

Apesar de uma liderança inicial dos EUA, a propriedade e uso de telefones celulares em A Europa agora é muito maior do que nos Estados Unidos. Ter um único sistema que funciona em qualquer

onde na Europa e com qualquer provedor é parte do motivo, mas há mais. UMA a segunda área em que os EUA e a Europa diferiam é na humilde questão do telefone números. Nos EUA, os telefones celulares são combinados com os telefones normais (fixos).

Assim, não há como um chamador ver se, digamos, (212) 234-5678 é um telefone fixo telefone (chamada barata ou gratuita) ou um telefone móvel (chamada cara). Para manter as pessoas de ficar nervosa ao fazer chamadas, as companhias telefônicas decidiram fazer o proprietário do telefone móvel pagar pelas chamadas recebidas. Como consequência, muitos as pessoas hesitaram em comprar um telefone celular por medo de incorrer em uma grande conta apenas

ceiving chamadas. Na Europa, os números de telefone celular têm um código de área especial (análogo)

gous para 800 e 900 números) para que sejam instantaneamente reconhecíveis. Consequentemente, a regra usual de " quem chama paga " também se aplica a telefones celulares na Europa (exceto para chamadas internacionais, onde os custos são divididos).

Um terceiro problema que teve um grande impacto na adoção é o uso generalizado de telefones celulares pré-pagos na Europa (até 75% em algumas áreas). Estes podem ser pur-perseguido em muitas lojas sem mais formalidade do que comprar uma câmera digital. Você pagar e você vai. Eles são pré-carregados com um saldo de, por exemplo, 20 ou 50 euros e pode ser recarregado (usando um código PIN secreto) quando o saldo cair para zero. Como consequência, praticamente todo adolescente e muitas crianças pequenas em A Europa tem telefones celulares (geralmente pré-pagos) para que seus pais possam localizá-los, sem o perigo de a criança subir uma conta enorme. Se o telefone celular for usado apenas ocasionalmente, seu uso é essencialmente gratuito, pois não há cobrança mensal ou cobrar pelas chamadas recebidas.

2.7.1 Telefones celulares de primeira geração (1G): voz analógica

Chega de falar sobre os aspectos políticos e de marketing dos telefones celulares. Agora deixe vamos olhar para a tecnologia, começando com o sistema mais antigo. Radiotele móvel

telefones foram usados esporadicamente para comunicação marítima e militar durante as primeiras décadas do século XX. Em 1946, o primeiro sistema para tele-carro telefones foi instalado em St. Louis. Este sistema usava um único transmissor grande no topo de um prédio alto e tinha um único canal, usado para enviar e receber. Para falar, o usuário precisava apertar um botão que habilitava o transmissor e desabilitava o receptor. Tais sistemas, conhecidos como sistemas **push-to-talk**, foram instalados em vários cidades começando no final dos anos 1950. Rádio CB, táxis e carros de polícia costumam usar este tecnologia. Na década de 1960, o **IMTS (Improved Mobile Telephone System)** foi instalado. Ele também usava um transmissor de alta potência (200 watts) no topo de uma colina, mas tinha dois frequências, uma para enviar e outra para receber, então o botão push-to-talk foi

Página 191

SEC. 2,7
O SISTEMA DE TELEFONE MÓVEL

167

não é mais necessário. Uma vez que todas as comunicações dos telefones móveis foram de entrada em um canal diferente dos sinais de saída, os usuários móveis poderiam não se ouvem (ao contrário do sistema push-to-talk usado nos táxis).

IMTS suportou 23 canais espalhados de 150 MHz a 450 MHz. Devido a o pequeno número de canais, os usuários muitas vezes tinham que esperar muito tempo antes de obter um

tom de discagem. Além disso, devido à grande potência dos transmissores do topo da colina, sistemas adjacentes

Os tems deveriam estar separados por várias centenas de quilômetros para evitar interferência. Contudo,

a capacidade limitada tornava o sistema impraticável.

Sistema de telefonia móvel avançado

Tudo isso mudou com **AMPS (Advanced Mobile Phone System)**, inventado pela Bell Labs e instalado pela primeira vez nos Estados Unidos em 1982. Ele também foi usado em Na Inglaterra, onde foi denominado TACS, e no Japão, onde foi denominado MCS-L1.

AMPS foi formalmente aposentado em 2008, mas vamos olhar para isso para entender o texto para os sistemas 2G e 3G que melhoraram nele.

Em todos os sistemas de telefonia móvel, uma região geográfica é dividida em **células**, é por isso que os dispositivos às vezes são chamados de telefones celulares. No AMPS, as células têm tipicamente de 10 a 20 km de diâmetro; em sistemas digitais, as células são menores. Cada a célula usa algum conjunto de frequências não usado por nenhum de seus vizinhos. A ideia chave que dá aos sistemas celulares muito mais capacidade do que os sistemas anteriores é o uso de células relativamente pequenas e a reutilização de frequências de transmissão nas proximidades (mas não

adjacentes) células. Considerando que um sistema IMTS com 100 km de diâmetro pode ter apenas uma chamada

em cada frequência, um sistema AMPS pode ter 100 células de 10 km na mesma área e ser capaz de ter de 10 a 15 chamadas em cada frequência, em células amplamente separadas.

Assim, o projeto celular aumenta a capacidade do sistema em pelo menos uma ordem de magnitude, mais à medida que as células ficam menores. Além disso, células menores significam que

menos energia é necessária, o que leva a transmissores menores e mais baratos e aparelhos.

A ideia de reutilização de frequência é ilustrada na Figura 2.45 (a). As células não são mais ou menos circulares, mas são mais fáceis de modelar como hexágonos. Na Fig. 2-45 (a), as células são todas do mesmo tamanho. Eles são agrupados em unidades de sete células. Cada letra indica um grupo de frequências. Observe que para cada conjunto de frequência, há um buffer com cerca de duas células de largura, onde essa frequência não é reutilizada, proporcionando

boa separação e baixa interferência.

Encontrar locais no ar para colocar antenas de estação base é um importante

questão. Este problema tem levado algumas operadoras de telecomunicações a formarem alianças com a Igreja Católica Romana, uma vez que esta última possui um número substancial de locais de antenas em potencial exaltado em todo o mundo, todos convenientemente sob um único homem agement.

Em uma área onde o número de usuários cresceu a ponto de o sistema ser sobrecarregado, a potência pode ser reduzida e as células sobrecarregadas divididas em menores

Página 192

168

A CAMADA FÍSICA

INDIVÍDUO. 2

G
F
UMA
B
C
D
E
G
F
UMA
B
C
D
E
G
F
UMA
B
C
D
E
(uma)
(b)

Figura 2-45. (a) As frequências não são reutilizadas em células adjacentes. (b) Para adicionar mais usuários, células menores podem ser usadas.

microcélulas para permitir maior reutilização de frequência, conforme mostrado na Figura 2.45 (b).

Telefone

as empresas às vezes criam microcélulas temporárias, usando torres portáteis com links de satélite em eventos esportivos, shows de rock e outros lugares onde um grande número bers de usuários móveis se reúnem por algumas horas.

No centro de cada célula está uma estação base para a qual todos os telefones do transmissão celular. A estação base consiste em um computador e transmissor / receptor conectado a uma antena. Em um sistema pequeno, todas as estações base estão conectadas a um único dispositivo denominado **MSC** (**Mobile Switching Center**) ou **MTSO** (**Mobile Central de comutação telefônica**). Em um maior, vários MSCs podem ser necessários, todos dos quais estão conectados a um MSC de segundo nível e assim por diante. Os MSCs são essenciais oficialmente terminais, como no sistema telefônico, e estão de fato conectados a pelo menos uma estação final do sistema telefônico. Os MSCs se comunicam com as estações base, uns aos outros e ao PSTN usando uma rede de comutação de pacotes.

A qualquer momento, cada telefone móvel está logicamente em uma célula específica e der o controle da estação base daquela célula. Quando um telefone celular fisicamente sai de uma célula, sua estação base percebe que o sinal do telefone está enfraquecendo e pergunta todas as estações base vizinhas quanta energia estão recebendo dela. Quando as respostas voltam, a estação base, em seguida, transfere a propriedade para o celular getting o sinal mais forte; sob a maioria das condições, essa é a célula onde o telefone agora está localizado. O telefone é então informado de seu novo chefe, e se um chamada está em andamento, é solicitada a mudança para um novo canal (porque o antigo está não reutilizado em nenhuma das células adjacentes). Este processo, chamado **handoff**, leva cerca de 300 msec. A atribuição do canal é feita pelo MSC, o centro nervoso do sistema tem. As estações base são, na verdade, apenas relés de rádio burros.

Página 193

169**Canais**

AMPS usa FDM para separar os canais. O sistema usa 832 full-duplex canais, cada um consistindo de um par de canais simplex. Este arranjo é conhecido como **FDD** (Frequency Division Duplex). Os 832 canais simplex de 824 a 849 MHz são usados para transmissão móvel para estação base e 832 simplex canais de 869 a 894 MHz são usados para transmissão de estação base para celular. Cada um desses canais simplex tem 30 kHz de largura.

Os 832 canais são divididos em quatro categorias. Canais de controle (base para móvel) são usados para gerenciar o sistema. Alerta de canais de paging (base para celular) usuários móveis para chamadas para eles. Canais de acesso (bidirecionais) são usados para chamadas configuração e atribuição de canal. Finalmente, os canais de dados (bidirecionais) transportam voz, fax ou dados. Uma vez que as mesmas frequências não podem ser reutilizadas em células próximas e

21

canais são reservados em cada célula para controle, o número real de canais de voz disponível por célula é muito menor do que 832, normalmente cerca de 45.

Gerenciamento de Chamadas

Cada telefone celular no AMPS tem um número de série de 32 bits e um número de 10 dígitos número de telefone em sua memória somente leitura programável. O número de telefone é representado como um código de área de 3 dígitos em 10 bits e um número de assinante de 7 dígitos em

24 bits. Quando um telefone é ligado, ele verifica uma lista pré-programada de 21 controles canais para encontrar o sinal mais poderoso. O telefone, então, transmite sua mensagem de 32 bits número de série e número de telefone de 34 bits. Como todas as informações de controle em AMPS, este pacote é enviado em formato digital, várias vezes, e com um erro-cor-código de correção, mesmo que os próprios canais de voz sejam analógicos.

Quando a estação base ouve o anúncio, ela diz ao MSC, que registra a existência de seu novo cliente e também informa a casa do cliente MSC de sua localização atual. Durante a operação normal, o telefone móvel registra-se novamente uma vez a cada 15 minutos.

Para fazer uma chamada, um usuário móvel liga o telefone e insere o número a ser chamado no teclado e pressiona o botão ENVIAR. O telefone então transmite o número a ser chamado e identidade própria no canal de acesso. Se uma colisão ocorrer maldito aí, ele tenta novamente mais tarde. Quando a estação base recebe o pedido, ela informa o MSC. Se o chamador for um cliente da empresa MSC (ou um de seus parceiros), o MSC procura um canal livre para a chamada. Se um for encontrado, o número do canal é enviado de volta ao canal de controle. O celular então se automuda automaticamente para o canal de voz selecionado e espera até a parte chamada atende o telefone.

As chamadas recebidas funcionam de maneira diferente. Para começar, todos os telefones inativos continuamente ouça o canal de paging para detectar mensagens direcionadas a eles. Quando uma chamada é colocada em um telefone móvel (de um telefone fixo ou outro telefone móvel), um pacote é enviado ao MSC local do receptor para descobrir onde ele está. Um pacote é então

170**A CAMADA FÍSICA****INDIVÍDUO. 2**

enviado para a estação base em sua célula atual, que envia uma transmissão no paging canal do formulário " Unidade 14, você está aí? " O telefone chamado responde com um " Sim " no canal de acesso. A base então diz algo como: " Unidade 14, ligue para você no canal 3. " Neste ponto, o telefone chamado muda para o canal 3 e começa a fazer sons de toque (ou tocar alguma melodia que o proprietário deu como um presente de aniversário).

2.7.2 Telefones móveis de segunda geração (2G): voz digital

A primeira geração de telefones celulares era analógica; a segunda geração é digital. Mudar para o digital tem várias vantagens. Ele fornece ganhos de capacidade por permitindo que os sinais de voz sejam digitalizados e comprimidos. Além disso, melhora a segurança por al-

baixo sinais de voz e controle a serem criptografados. Isso, por sua vez, impede a fraude e espionagem, seja por varredura intencional ou ecos de outras chamadas devido a Propagação de RF. Finalmente, permite novos serviços, como mensagens de texto.

Assim como não houve padronização mundial durante a primeira geração, também não houve padronização mundial durante o segundo. De várias diferentes sistemas foram desenvolvidos e três foram amplamente implantados. D-

AMPS (Digital Advanced Mobile Phone System) é uma versão digital do AMPS que coexiste com AMPS e usa TDM para fazer várias chamadas na mesma frequência canal de freqüência. É descrito no International Standard IS-54 e seu sucessor IS-136. **GSM (Sistema Global para Comunicações Móveis)** surgiu como o sistema dominante, e embora tenha demorado para pegar nos EUA, agora é usado por em todos os lugares do mundo. Como o D-AMPS, o GSM é baseado em uma mistura de FDM e TDM. **CDMA (Code Division Multiple Access),** descrito em **International**

O padrão IS-95 é um tipo de sistema completamente diferente e não se baseia em nenhum FDM nem TDM. Embora o CDMA não tenha se tornado o sistema 2G dominante, seu a tecnologia tornou-se a base para os sistemas 3G.

Além disso, o nome **PCS (Personal Communications Services)** às vezes é usado na literatura de marketing para indicar um sistema de segunda geração (ou seja, digital) tem. Originalmente, significava um telefone móvel usando a banda de 1900 MHz, mas isso dis- a tintura raramente é feita agora.

Descreveremos agora o GSM, uma vez que é o sistema 2G dominante. Na próxima Na seção, teremos mais a dizer sobre CDMA quando descrevermos os sistemas 3G.

GSM - O Sistema Global para Comunicações Móveis

O GSM começou a vida na década de 1980 como um esforço para produzir um único 2G europeu padrão. A tarefa foi atribuída a um grupo de telecomunicações chamado (em francês) Groupe Specialé Mobile. Os primeiros sistemas GSM foram implantados a partir de 1991 e foram um sucesso rápido. Logo ficou claro que o GSM seria mais do que um sucesso europeu, com aceitação se estendendo a países tão distantes como Austrália, então GSM foi renomeado para ter um apelo mais mundial.

Página 195

SEC. 2,7

O SISTEMA DE TELEFONE MÓVEL

171

GSM e outros sistemas de telefonia móvel que estudaremos reter de sistemas 1G tem um design baseado em células, reutilização de frequência entre células e mobilidade com transferências conforme os assinantes mudam. São os detalhes que diferem. Aqui, vamos brevemente

discutir algumas das principais propriedades do GSM. No entanto, o padrão GSM impresso dard tem mais de 5.000 [sic] páginas. Uma grande fração deste material se relaciona com aspectos de engenharia do sistema, especialmente o projeto de receptores para lidar com propagação de sinal tipath e sincronização de transmissores e receptores. Nenhum isso será até mencionado aqui.

A Fig. 2-46 mostra que a arquitetura GSM é semelhante à arquitetura AMPS-, embora os componentes tenham nomes diferentes. O próprio celular agora está fornecido no aparelho e um chip removível com assinante e informações de conta mação chamada **cartão SIM**, abreviação de **Módulo de Identidade do Assinante**. É o SIM cartão que ativa o aparelho e contém segredos que permitem ao celular e à rede trabalho se identifica e criptografa conversas. Um cartão SIM pode ser removido e conectado a um aparelho diferente para transformá-lo em seu celular até a rede está em causa.

VLR

MSC
Ar
interface
Torre de celular e
estação base
PSTN
SIM
cartão
Aparelho portátil
HLR
BSC
BSC

Figura 2-46. Arquitetura de rede móvel GSM.

O celular se comunica com as estações base de celular por meio de uma **interface aérea** que iremos de-

escreba em um momento. As estações base de células são conectadas a um **BSC (Base**

Controlador de estação) que controla os recursos de rádio das células e lida com o handoff.

O BSC, por sua vez, é conectado a um MSC (como no AMPS) que roteia chamadas e con-
necta-se à PSTN (Rede Telefônica Pública Comutada).

Para poder rotear chamadas, o MSC precisa saber onde os celulares podem atualmente ser encontrado. Ele mantém um banco de dados de celulares próximos que estão associados ao céluas que ele gerencia. Este banco de dados é denominado **VLR (Visitor Location Register)**. Também existe um banco de dados na rede móvel que fornece a última localização conhecida de cada celular. É denominado **HLR (Home Location Register)**. Este banco de dados é usado para rotear as chamadas recebidas para os locais certos. Ambos os bancos de dados devem ser mantidos

atualizado conforme os celulares se movem de uma célula para outra.

Vamos agora descrever a interface aérea com alguns detalhes. GSM funciona em um intervalo de frequências em todo o mundo, incluindo 900, 1800 e 1900 MHz. Mais espectro é alocado do que para AMPS, a fim de oferecer suporte a um número muito maior de usuários. GSM

Página 196

172

A CAMADA FÍSICA INDIVÍDUO. 2

é um sistema celular duplex de divisão de frequência, como AMPS. Ou seja, cada celular transmite em uma frequência e recebe em outra, frequência mais alta (55 MHz superior para GSM versus 80 MHz superior para AMPS). No entanto, ao contrário do AMPS, com o GSM, um único par de frequência é dividido por multiplexação por divisão de tempo no tempo

slots. Desta forma, ele é compartilhado por vários celulares.

Para lidar com vários celulares, os canais GSM são muito mais amplos que o AMPS canais (200 kHz versus 30 kHz). Um canal de 200 kHz é mostrado na Figura 2-47.

Um sistema GSM operando na região de 900 MHz tem 124 pares de canais simplex nels. Cada canal simplex tem 200 kHz de largura e suporta oito canais separados conexões nele, usando multiplexação por divisão de tempo. Cada estação atualmente ativa é atribuído um intervalo de tempo em um par de canais. Teoricamente, 992 canais podem ser suportados em cada célula, mas muitos deles não estão disponíveis, para evitar a frequência conflitos com células vizinhas. Na Figura 2-47, os oito intervalos de tempo sombreados serão todos longo para a mesma conexão, quatro deles em cada direção. Transmitindo e re-
o recebimento não acontece no mesmo intervalo de tempo porque os rádios GSM não podem transmitir e receber ao mesmo tempo e leva tempo para mudar de um para o de outros. Se o dispositivo móvel atribuído a 890,4 / 935,4 MHz e o intervalo de tempo 2 desejado para transmitir para a estação base, ele usaria os quatro slots sombreados inferiores (e o aqueles que os seguem no tempo), colocando alguns dados em cada slot até que todos os dados tenham

foi enviado.

959,8 MHz
935,4 MHz
935,2 MHz
914,8 MHz
890,4 MHz
890,2 MHz

```

Frequência
Base
para celular
Móvel
basear
124
2
1
124
2
1
Canal
Quadro TDM
Tempo

```

Figura 2-47. O GSM usa 124 canais de frequência, cada um dos quais usa um oito sistema de slot TDM.

Os slots TDM mostrados na Figura 2.47 fazem parte de uma hierarquia de enquadramento complexa. Cada slot TDM tem uma estrutura específica e grupos de slots TDM formam tiframes, também com uma estrutura específica. Uma versão simplificada desta hierarquia é mostrado na Figura 2-48. Aqui podemos ver que cada slot TDM consiste em um compartimento de 148 bits quadro de dados que ocupa o canal por 577 µs (incluindo um tempo de guarda de 30 µs

Página 197

SEC. 2,7
O SISTEMA DE TELEFONE MÓVEL

173

após cada slot). Cada quadro de dados começa e termina com três bits 0, para quadro del-fins de formação. Ele também contém dois campos de *informação* de 57 bits , cada um tendo um bit de controle que indica se o seguinte campo de *informações* é para voz ou dados. Entre os campos de *informações* está um campo de *sincronização* (treinamento) de 26 bits que é usado

pelo receptor para sincronizar com os limites do quadro do remetente.

```

C
T
eu
0
1
2
3
4
5
6
7
8
9 10 11
13 14 15 16 17 18 19 20 21 22 23 24
Multiframe de 32.500 bits enviado em 120 ms
0
1
2
3
4
5
6
7
Quadro TDM de 1250 bits enviado em 4,615 mseg
8,25 bits
(30 µseg)
tempo de guarda
Reservado
para o futuro
usar
000
000
Em formação
Em formação
Sincronizar
Frame de dados de 148 bits enviado em 547 µseg
Bits
3
3
57
57
26
Bit de voz / dados

```

Figura 2-48. Uma parte da estrutura de enquadramento do GSM.

Um quadro de dados é transmitido em 547 µs, mas um transmissor só tem permissão para enviar um quadro de dados a cada 4,615 ms, uma vez que está compartilhando o canal com sete outras estações. A taxa bruta de cada canal é 270.833 bps, dividida em oito slots.

Comercial. No entanto, como acontece com o AMPS, a sobrecarga consome uma grande fração da banda

largura, deixando 24,7 kbps de carga útil por usuário antes de corrigir o erro de correção. Após a correção do erro, resta 13 kbps para a fala. Embora isso seja substancialmente menor que os 64 kbps PCM para sinais de voz não comprimidos no telefone fixo, a compressão no dispositivo móvel pode atingir esses níveis com little perda de qualidade.

Como pode ser visto na Figura 2.48, oito quadros de dados constituem um quadro TDM e 26 quadros TDM formam um multiframe de 120 ms. Dos 26 quadros TDM em um multiframe, o slot 12 é usado para controle e o slot 25 é reservado para uso futuro, então apenas 24 estão disponíveis para tráfego de usuários.

No entanto, além do multiframe de 26 slots mostrado na Figura 2-48, um slot de 51 slots (não mostrado) também é usado. Alguns desses slots são usados para armazenar vários canais de controle usados para gerenciar o sistema. O **canal de controle de transmissão** é um fluxo contínuo de saída da estação base contendo os

identidade e o status do canal. Todas as estações móveis monitoram a intensidade do sinal para ver quando eles mudaram para uma nova célula.

Página 198

174

A CAMADA FÍSICA

INDIVÍDUO. 2

O **canal de controle dedicado** é usado para atualização de localização, registro, e configuração da chamada. Em particular, cada BSC mantém um banco de dados de estações móveis atualmente sob sua jurisdição, o VLR. Informações necessárias para manter o VLR são enviadas no canal de controle dedicado.

Por fim, existe o **canal de controle comum**, que é dividido em três subcanais lógicos. O primeiro desses subcanais é o **canal de paging**, que a estação base usa para anunciar chamadas recebidas. Cada estação móvel o monitora continuamente para observar as chamadas que deve atender. O segundo é o **ac- aleatório canal de acesso**, que permite aos usuários solicitar um slot no **canal de controle dedicado**. Se duas solicitações entrarem em conflito, elas serão distorcidas e deverão ser tentadas novamente mais tarde. Usando

o slot de canal de controle dedicado, a estação pode estabelecer uma chamada. O slot atribuído é anunciado no terceiro subcanal, o **canal de concessão de acesso**.

Finalmente, o GSM difere do AMPS na forma como o handoff é tratado. No AMPS, o MSC o gerencia completamente sem a ajuda dos dispositivos móveis. Com tempo slots no GSM, o celular não envia nem recebe a maior parte do tempo. Os slots ociosos são uma oportunidade para o celular medir a qualidade do sinal para outras estações base próximas. Ele faz isso e envia essas informações ao BSC. O BSC pode usá-lo para determinar quando um celular está saindo de uma célula e entrando em outra, ele pode realizar a transferência. Este design é denominado **MAHO (Mobile Assisted HandOff)**.

2.7.3 Telefones móveis de terceira geração (3G): voz digital e dados

A primeira geração de telefones celulares era de voz analógica, e a segunda geração era voz digital. A terceira geração de telefones celulares, ou **3G** como é chamado, tem tudo a ver com voz e dados digitais.

Vários fatores estão impulsionando o setor. Primeiro, o tráfego de dados já excede o tráfego de voz na rede fixa e está crescendo exponencialmente, enquanto o tráfego de voz é essencialmente plano. Muitos especialistas do setor esperam que o tráfego de dados domine

voz inata em dispositivos móveis também em breve. Em segundo lugar, o telefone, entretenimento,

e as indústrias de computadores se tornaram digitais e estão convergindo rapidamente. Muitos as pessoas estão babando sobre dispositivos portáteis leves que funcionam como um telefone, music e player de vídeo, terminal de e-mail, interface da Web, máquina de jogos e muito mais, tudo isso com conectividade mundial sem fio à Internet em alta largura de banda.

O iPhone da Apple é um bom exemplo desse tipo de dispositivo 3G. Com isso gente ficar viciado em serviços de dados sem fio, e os volumes de dados sem fio da AT&T estão aumentando

com a popularidade dos iPhones. O problema é que o iPhone usa uma rede de 2,5 G funcionar (uma rede 2G aprimorada, mas não uma verdadeira rede 3G) e não há capacidade de dados suficiente para manter os usuários satisfeitos. A telefonia móvel 3G tem tudo a ver com fornecendo largura de banda sem fio suficiente para manter esses futuros usuários felizes.

A ITU tentou ser um pouco mais específica sobre essa visão, começando por 1992. Ele emitiu um plano para chegar lá chamado **IMT-2000**, onde IMT estava

Página 199

SEC. 2,7

O SISTEMA DE TELEFONE MÓVEL

175

para **telecomunicações móveis internacionais**. Os serviços básicos que o

A rede IMT-2000 deveria fornecer aos seus usuários:

1. Transmissão de voz de alta qualidade.
2. Mensagens (substituindo e-mail, fax, SMS, chat, etc.).
3. Multimídia (reprodução de música, exibição de vídeos, filmes, televisão, etc.).
4. Acesso à Internet (navegação na Web, incluindo páginas com áudio e vídeo).

Os serviços adicionais podem ser videoconferência, telepresença, jogos em grupo - e m-commerce (acenando com o telefone no caixa para pagar em uma loja).

Além disso, todos esses serviços devem estar disponíveis em todo o mundo (com conexão automática via satélite quando nenhuma rede terrestre pode ser localizada), instantaneamente (sempre ligado) e com garantias de qualidade de serviço.

A ITU imaginou uma única tecnologia mundial para IMT-2000, portanto, os usuários poderiam construir um único dispositivo que poderia ser vendido e usado em qualquer lugar do

mundo (como CD players e computadores e ao contrário de telefones celulares e televisores).

Ter uma única tecnologia também tornaria a vida muito mais simples para as operações de rede torres e incentivaria mais pessoas a usar os serviços. Guerras de formato, como a batalha entre Betamax e VHS com videocassetes não são bons para os negócios.

No final das contas, isso foi um pouco otimista. O número 2.000 representou três coisas: (1) o ano em que deveria entrar em serviço, (2) a frequência com que deve operar em (em MHz), e (3) a largura de banda que o serviço deve ter (em kbps). Não foi bem-sucedido em nenhuma das três acusações. Nada foi implementado em 2000. A ITU recomendou que todos os governos reservassem espectro em 2 GHz para que os dispositivos possam se movimentar sem problemas de um país para outro. China reservada

a largura de banda necessária, mas ninguém mais fez. Finalmente, foi reconhecido que 2 Atualmente, o Mbps não é viável para usuários *muito* móveis (devido à dificuldade de realizar transferências com rapidez suficiente). Mais realista é 2 Mbps para estacionário usuários internos (que irão competir de frente com ADSL), 384 kbps para pessoas que andam e 144 kbps para conexões em carros.

Apesar desses contratemplos iniciais, muito foi conquistado desde então. Several propostas de IMT foram feitas e, após alguma seleção, caiu para duas principais. O primeiro, **WCDMA** (**Wideband CDMA**), foi proposto por Ericsson e foi empurrado pela União Europeia, que o chamou de **UMTS** (**Universo sal Sistema de telecomunicações móveis**). O outro contendor era **CDMA2000**, proposto pela Qualcomm.

Ambos os sistemas são mais semelhantes do que diferentes, pois se baseiam em banda larga CDMA; WCDMA usa canais de 5 MHz e CDMA2000 usa 1,25-

Canais de MHz. Se os engenheiros da Ericsson e da Qualcomm fossem colocados em uma sala e instruídos a chegar a um projeto comum, eles provavelmente conseguiriam encontrar um rapidamente.

O problema é que o verdadeiro problema não é a engenharia, mas a política (como sempre). A Europa queria um sistema que interagisse com o GSM, enquanto os EUA queriam um

Página 200

176

A CAMADA FÍSICA
INDIVÍDUO. 2

sistema que era compatível com um já amplamente implantado nos EUA (IS-95).

Cada lado também apoiou sua empresa local (a Ericsson está sediada na Suécia; Qualcomm está na Califórnia). Finalmente, a Ericsson e a Qualcomm estiveram envolvidas em vários diversas ações judiciais sobre suas respectivas patentes CDMA.

Em todo o mundo, 10–15% dos assinantes móveis já usam tecnologias 3G. No América do Norte e Europa, cerca de um terço dos assinantes móveis são 3G. Japão foi um dos primeiros a adotar e agora quase todos os telefones celulares no Japão são 3G. Estes os números incluem a implantação de UMTS e CDMA2000, e 3G continua para ser um grande caldeirão de atividade enquanto o mercado balança. Para adicionar à confusão Sion, UMTS tornou-se um único padrão 3G com múltiplas opções incompatíveis, incluindo CDMA2000. Essa mudança foi um esforço para unificar os vários campos, mas apenas documentos sobre as diferenças técnicas e obscurece o foco de esforços. Usaremos UMTS para significar WCDMA, diferente de CDMA2000.

Focaremos nossa discussão no uso de CDMA em redes celulares, já que é a característica distintiva de ambos os sistemas. CDMA não é FDM nem TDM mas uma espécie de mix em que cada usuário envia na mesma banda de frequência no mesmo tempo. Quando foi proposto pela primeira vez para sistemas celulares, a indústria o deu aproximadamente a mesma reação que Colombo teve pela primeira vez da Rainha Isabel quando ele propôs chegar à Índia navegando na direção errada. Contudo, por meio da persistência de uma única empresa, a Qualcomm, a CDMA teve sucesso como um Sistema 2G (IS-95) e amadureceu a ponto de se tornar a base técnica para 3G.

Para fazer o CDMA funcionar na configuração do telefone móvel, é necessário mais do que o técnica CDMA básica que descrevemos na seção anterior. Especificamente, descrevemos CDMA síncrono, em que as sequências de chips são exatamente ortogonal. Este design funciona quando todos os usuários estão sincronizados na hora de início de suas sequências de chips, como no caso da estação base transmitindo para celulares.

A estação base pode transmitir as sequências de chips começando ao mesmo tempo para que os sinais serão ortogonais e poderão ser separados. No entanto, é difícil sincronizar as transmissões de telefones móveis independentes. Sem cuidado, seu as transmissões chegariam na estação base em horários diferentes, sem garantia de ortogonalidade. Para permitir que os celulares enviem para a estação base sem sincronização, queremos sequências de código que sejam ortogonais entre si em todos os deslocamentos possíveis, não simplesmente quando eles estão alinhados no início.

Embora não seja possível encontrar sequências que sejam exatamente ortogonais para este caso geral, longas sequências pseudo-aleatórias chegam perto o suficiente. Eles têm a propriedade que, com alta probabilidade, eles têm uma baixa **correlação cruzada** com cada outro em todos os deslocamentos. Isso significa que quando uma sequência é multiplicada por outra sequência e somados para calcular o produto interno, o resultado será pequeno; isto seria zero se fossem ortogonais. (Intuitivamente, as sequências aleatórias devem sempre maneiras parecem diferentes umas das outras. Multiplicá-los juntos deve então produzir um sinal aleatório, que somará um pequeno resultado.) Isso permite que um receptor filtre transmissões indesejadas fora do sinal recebido. Além disso, a **autocorrelação** de

Página 201

SEC. 2,7

sequências pseudo-aleatórias também são pequenas, com alta probabilidade, exceto em um ponto zero

conjunto. Isso significa que quando uma sequência é multiplicada por uma cópia atrasada de si mesma

e somados, o resultado será pequeno, exceto quando o atraso for zero. (Intuitivamente, uma sequência aleatória atrasada parece uma sequência aleatória diferente, e nós estamos de volta ao caso de correlação cruzada.) Isso permite que um receptor bloqueeie no início de a transmissão desejada no sinal recebido.

O uso de sequências pseudo-aleatórias permite que a estação base receba mensagens CDMA sábios de celulares não sincronizados. No entanto, uma suposição implícita em nossa discussão do CDMA é que os níveis de potência de todos os celulares são os mesmos na receiver. Se não forem, uma pequena correlação cruzada com um sinal poderoso pode sobrecarregar uma grande autocorrelação com um sinal fraco. Assim, a potência de transmissão em dispositivos móveis deve ser controlado para minimizar a interferência entre sinais concorrentes nais. É essa interferência que limita a capacidade dos sistemas CDMA.

Os níveis de potência recebidos em uma estação base dependem de quanto longe os transmitters são, bem como a quantidade de energia que transmitem. Pode haver muitos dispositivos móveis

estações em distâncias variáveis da estação base. Uma boa heurística para equalizar a potência recebida é para cada estação móvel transmitir para a estação base no inverso do nível de potência que recebe da estação base. Em outras palavras, um estação móvel que recebe um sinal fraco da estação base usará mais energia do que receber um sinal forte. Para mais precisão, a estação base também oferece cada feedback móvel para aumentar, diminuir ou manter estável sua potência de transmissão. O feedback é frequente (1500 vezes por segundo) porque um bom controle de potência é importante para minimizar a interferência.

Outra melhoria em relação ao esquema CDMA básico que descrevemos anteriormente é permitir que diferentes usuários enviem dados em taxas diferentes. Este truque é realizado naturalmente em CDMA, fixando a taxa na qual os chips são transmitidos e atribuindo usuários de sequências de chips de diferentes comprimentos. Por exemplo, em WCDMA, a taxa de chip

é 3,84 Mchips / seg e os códigos de espalhamento variam de 4 a 256 chips. Com um 256-código do chip, cerca de 12 kbps é deixado após a correção de erros, e esta capacidade é suficiente para uma chamada de voz. Com um código de 4 chips, a taxa de dados do usuário é próxima a 1 Mbps.

Códigos de comprimento intermediário fornecem taxas intermediárias; para obter vários Mbps, o celular deve usar mais de um canal de 5 MHz ao mesmo tempo.

Agora vamos descrever as vantagens do CDMA, uma vez que tratamos os problemas de fazê-lo funcionar. Tem três vantagens principais. Primeiro, CDMA pode melhorar a capacidade, aproveitando pequenos períodos quando os mitters estão em silêncio. Em chamadas de voz educadas, uma parte fica em silêncio enquanto a outra fala. Em

em média, a linha fica ocupada apenas 40% do tempo. No entanto, as pausas podem ser pequenas e são difíceis de prever. Com sistemas TDM ou FDM, não é possível reajustar assinar slots de tempo ou canais de frequência com rapidez suficiente para se beneficiar desses pequenos

silêncios. No entanto, em CDMA, simplesmente não transmitindo a um usuário diminui o referência para outros usuários, e é provável que alguma fração dos usuários não seja transmitindo em uma célula ocupada a qualquer momento. Assim, o CDMA tira vantagem do excessos esperados para permitir um maior número de chamadas simultâneas.

Em segundo lugar, com o CDMA, cada célula usa as mesmas frequências. Ao contrário de GSM e AMPS, FDM não são necessários para separar as transmissões de diferentes usuários. Isto elimina tarefas complicadas de planejamento de frequência e melhora a capacidade. Isso também torna mais fácil para uma estação base usar várias antenas direcionais ou **setoriais** **antenas**, em vez de uma antena omnidirecional. Concentração de antenas direcionais trate um sinal na direção pretendida e reduza o sinal e, portanto, entre referência, em outras direções. Isso, por sua vez, aumenta a capacidade. Três projetos de setor são comuns. A estação base deve rastrear o celular conforme ele se move do setor para setor. Esse rastreamento é fácil com CDMA porque todas as frequências são usadas em todos setores.

Terceiro, o CDMA facilita a **transferência suave**, em que o celular é adquirido pelo nova estação base antes que a anterior termine. Desta forma, não há perda de continuidade. A transferência suave é mostrada na Figura 2-49. É fácil com CDMA porque todas as frequências são usadas em cada célula. A alternativa é um **handoff difícil**, em que o a estação base antiga desconecta a chamada antes que a nova a adquira. Se o novo for incapaz de adquiri-lo (por exemplo, porque não há frequência disponível), a chamada é desconectado abruptamente. Os usuários tendem a notar isso, mas ocasionalmente é inevitável com o design atual. Handoff difícil é a norma com projetos FDM para evitar o custo de ter o celular transmitindo ou recebendo em duas frequências simultaneamente.

(uma)
(b)
(c)

Figura 2-49. Transferência suave (a) antes, (b) durante e (c) depois.

Muito tem sido escrito sobre 3G, a maior parte elogiando-o como a melhor coisa desde o pão frito. Enquanto isso, muitas operadoras tomaram medidas cautelosas no direção de 3G, indo para o que é às vezes chamado **2.5G**, embora poderia ser 2,1 g ser mais preciso. Um desses sistemas é **EDGE (taxas de dados aprimoradas para GSM Evolution)**, que é apenas GSM com mais bits por símbolo. O problema é que mais bits por símbolo também significam mais erros por símbolo, então EDGE tem nove diferentes esquemas de modulação e correção de erros, diferindo em termos de quanto a largura de banda é dedicada a corrigir os erros introduzidos pela velocidade mais alta. EDGE é um passo ao longo de um caminho evolutivo que é definido de GSM para WCDMA. Da mesma forma, há um caminho evolutivo definido para os operadores para atualização de redes IS-95 para CDMA2000.

Mesmo que as redes 3G não estejam totalmente implantadas ainda, alguns pesquisadores consideram 3G como um negócio fechado. Essas pessoas já estão trabalhando em sistemas 4G sob o

SEC. 2,7
O SISTEMA DE TELEFONE MÓVEL
179

nome do **LTE (Long Term Evolution)**. Alguns dos recursos propostos do 4G em clude: alta largura de banda; onipresença (conectividade em todos os lugares); integração perfeita com outras redes IP com e sem fio, incluindo pontos de acesso 802.11; adaptação ativa de recursos e espectro; e alta qualidade de serviço para muitos meios de comunicação. Para obter mais informações, consulte Astely et al. (2009) e Larmo et al. (2009).

Enquanto isso, as redes sem fio com níveis de desempenho 4G já estão acessível. O principal exemplo é o **802.16**, também conhecido como **WiMAX**. Para uma visão geral de WiMAX móvel, ver Ahmadi (2009). Dizer que a indústria está em um estado de mudança é um eufemismo enorme. Verifique novamente em alguns anos para ver o que aconteceu.

2.8 TELEVISÃO A CABO

Já estudamos os sistemas de telefonia fixa e sem fio em uma feira quantidade de detalhes. Ambos irão claramente desempenhar um papel importante nas redes futuras. Mas há outro jogador importante que surgiu na última década para a Internet acesso: redes de televisão por cabo. Muitas pessoas hoje em dia pegam seus telefones e

Serviço de Internet por cabo. Nas seções a seguir, veremos a televisão a cabo visão como uma rede em mais detalhes e contrastá-la com os sistemas de telefonia que acabei de estudar. Algumas referências relevantes para mais informações são Donaldson e Jones (2001), Dutta-Roy (2001) e Fellows e Jones (2001).

2.8.1 Antena de televisão comunitária

A televisão a cabo foi concebida no final dos anos 1940 como uma forma de fornecer melhores acolhimento a pessoas que vivem em áreas rurais ou montanhosas. O sistema inicialmente consistia de uma grande antena no topo de uma colina para capturar o sinal de televisão do ar, um amplificador, chamado **headend**, para fortalecer o sinal, e um cabo coaxial para entregá-lo às casas das pessoas, conforme ilustrado na Figura 2-50.

Toque
Cabo coaxial
Drop cabo
Headend
Antena para pegar
sinais distantes

Figura 2-50. Um dos primeiros sistemas de televisão a cabo.

Nos primeiros anos, a televisão a cabo era chamada de **Community Antenna Television**. Foi uma operação muito familiar; qualquer pessoa hábil com eletrônica

Página 204

180

A CAMADA FÍSICA INDIVÍDUO. 2

poderia criar um serviço para sua cidade, e os usuários contribuiriam para pagar os custos. Conforme o número de assinantes crescia, cabos adicionais eram emendados na origem cabo final e amplificadores foram adicionados conforme necessário. A transmissão era unilateral, de o headend para os usuários. Em 1970, milhares de sistemas independentes existiam. Em 1974, a Time Inc. iniciou um novo canal, Home Box Office, com novo conteúdo (filmes) distribuído apenas por cabo. Outros canais somente a cabo seguiram, com foco sobre notícias, esportes, culinária e muitos outros tópicos. Este desenvolvimento deu origem a duas mudanças na indústria. Primeiro, as grandes corporações começaram a comprar sistemas de cabo e instalação de novos cabos para adquirir novos assinantes. Segundo, há era agora uma necessidade de conectar vários sistemas, muitas vezes em cidades distantes, a fim de dis-

homenagear os novos canais a cabo. As empresas de cabo começaram a instalar cabos entre as cidades para conectá-los todos em um único sistema. Este padrão era análogo a o que aconteceu na indústria de telefonia 80 anos antes com a conexão de estações finais anteriormente isoladas para possibilitar chamadas de longa distância.

2.8.2 Internet por cabo

Ao longo dos anos, o sistema de cabos cresceu e os cabos entre as várias cidades foram substituídos por fibra de alta largura de banda, semelhante ao que aconteceu em o sistema telefônico. Um sistema com fibra para os percursos de longa distância e coaxiais o cabo para as casas é chamado de sistema **HFC** (**Hybrid Fiber Coax**). O eletrônicos conversores ópticos que fazem interface entre as partes ópticas e elétricas do sistema tem são chamados de **nós de fibra**. Porque a largura de banda da fibra é muito maior do que o coaxial, um nó de fibra pode alimentar vários cabos coaxiais. Parte de um moderno O sistema HFC é mostrado na Figura 2-51 (a).

Na última década, muitas operadoras de cabo decidiram entrar na Internet negócios de acesso e, freqüentemente, negócios de telefonia também. Diferenças técnicas entre a planta de cabo e a planta de telefone teve um efeito sobre o que tinha que ser feito para atingir esses objetivos. Por um lado, todos os amplificadores unilaterais do sistema teve que ser substituído por amplificadores bidirecionais para suportar tanto a montante como a jusante

transmissões de fluxo. Enquanto isso acontecia, os primeiros sistemas de Internet por cabo tems usou a rede de televisão a cabo para transmissões downstream e um dial conexão ascendente através da rede telefônica para transmissões upstream. Foi um solução alternativa inteligente, mas não muito como uma rede em comparação com o que poderia ser.

No entanto, há outra diferença entre o sistema HFC da Figura 2-51 (a) e o sistema telefônico da Figura 2.51 (b), que é muito mais difícil de remover. Em baixo os bairros, um único cabo é compartilhado por muitas casas, enquanto na sistema telefônico, cada casa tem seu próprio loop local privado. Quando usado para televisão transmissão de íons, esse compartilhamento é um ajuste natural. Todos os programas são transmitidos em o cabo e não importa se há 10 ou 10.000 espectadores.

Quando o mesmo cabo é usado para acesso à Internet, no entanto, é muito importante se houver são 10 usuários ou 10.000. Se um usuário decidir baixar um arquivo muito grande, isso a largura de banda está potencialmente sendo retirada de outros usuários. Quanto mais usuários lá

Página 205

SEC. 2,8
TELEVISÃO À CABO

181

Cobre
par trançado
Interruptor
Pedágio
escritório
Cabeça-
fim
Alta largura de banda
tronco de fibra
Fim
escritório
Local
ciclo
(uma)
(b)
casa
Alta largura de banda
fibra
tronco
Coaxial
cabos
casa
Toque
Nó de fibra
Fibra
Fibra

Figura 2-51. (a) Televisão a cabo. (b) O sistema de telefonia fixa.

são, mais competição existe por largura de banda. O sistema telefônico não tem esta propriedade particular: baixar um arquivo grande em uma linha ADSL não reduza a largura de banda do seu vizinho. Por outro lado, a largura de banda do coaxial é muito maior do que os pares trançados, então você pode ter sorte se seus vizinhos o fizerem não uso muito a Internet.

A forma como a indústria de cabos enfrentou esse problema é dividir os cabos longos e conecte cada um diretamente a um nó de fibra. A largura de banda do headend para cada nó de fibra é efetivamente infinito, portanto, desde que não haja muitos sub escritores em cada segmento de cabo, a quantidade de tráfego é administrável. Típica

Página 206

182

A CAMADA FÍSICA
INDIVÍDUO. 2

cabos hoje em dia têm 500-2000 casas, mas à medida que mais e mais pessoas assinam à Internet por cabo, a carga pode ficar muito grande, exigindo mais divisão e mais nós de fibra.

2.8.3 Alocação de espectro

Tirar todos os canais de TV e usar a infraestrutura de cabo estritamente para acesso à Internet provavelmente geraria um bom número de clientes irados, então as empresas de cabo estão hesitantes em fazer isso. Além disso, a maioria das cidades regulamenta fortemente o que está no cabo, então as operadoras de cabo não teriam permissão para fazer isso, mesmo que

eles realmente queriam. Como consequência, eles precisavam encontrar uma maneira de ter televisão e Internet coexistem pacificamente no mesmo cabo.

A solução é construir a multiplexação por divisão de frequência. Televisão à cabo canais na América do Norte ocupam a região de 54-550 MHz (exceto para rádio FM, de 88 a 108 MHz). Esses canais têm 6 MHz de largura, incluindo bandas de guarda, e pode transportar um canal de televisão analógico tradicional ou vários canais de televisão digital canais. Na Europa, o limite inferior geralmente é de 65 MHz e os canais são de 6 a 8 MHz de largura para a resolução mais alta exigida por PAL e SECAM, mas de outra forma o esquema de alocação é semelhante. A parte baixa da banda não é usada. Moderno os cabos também podem operar bem acima de 550 MHz, freqüentemente em até 750 MHz ou mais. A solução escolhida foi introduzir canais upstream na banda de 5–42 MHz (um pouco mais alto na Europa) e use as frequências na extremidade alta para o baixo sinais de fluxo. O espectro do cabo é ilustrado na Figura 2-52.

0
108
televisão
televisão
Dados downstream
Frequências downstream
Rio acima
dados
Rio acima
frequências
FM
550
750 MHz
5 42 54 88

Figura 2-52. Alocação de frequência em um sistema típico de TV a cabo usado para Internet acesso à rede.

Observe que, uma vez que os sinais de televisão são todos downstream, é possível usar amplificadores upstream que funcionam apenas na região de 5–42 MHz e downstream amplificadores que funcionam apenas a 54 MHz e acima, conforme mostrado na figura. Assim, obtemos uma assimetria nas larguras de banda upstream e downstream porque mais especificações trum está disponível acima da televisão do que abaixo dela. Por outro lado, a maioria dos usuários querem mais tráfego downstream, então as operadoras de cabo não estão insatisfeitas com este fato

Página 207

SEC. 2,8
TELEVISÃO À CABO

183

da vida. Como vimos anteriormente, as companhias telefônicas geralmente oferecem um DSL assimétrico

serviço, mesmo que eles não tenham nenhuma razão técnica para fazê-lo.

Além de atualizar os amplificadores, a operadora deve atualizar os headend, também, de um amplificador burro para um sistema de computador digital inteligente com uma interface de fibra de alta largura de banda para um ISP. Freqüentemente, o nome é atualizado como

bem, de "headend" para **CMTS (Cable Modem Termination System)**. No a seguir, vamos nos abster de fazer uma atualização de nome e ficar com o tradicional "headend."

2.8.4 Modems a cabo

O acesso à Internet requer um modem a cabo, um dispositivo que possui duas interfaces: um para o computador e outro para a rede a cabo. Nos primeiros anos do cabo Internet, cada operadora tinha um modem a cabo proprietário, que foi instalado por um técnico da empresa de cabo. No entanto, logo ficou claro que um padrão aberto dard criaria um mercado competitivo de modem a cabo e reduziria os preços, portanto incentivo ao uso do serviço. Além disso, fazer os clientes comprarem cabo modems nas lojas e instalá-los eles próprios (como fazem com o acesso sem fio pontos) eliminaria as temidas rolagens de caminhão.

Consequentemente, as maiores operadoras de cabo se uniram a uma empresa chamada CableLabs para produzir um padrão de modem a cabo e para testar produtos para conformidade

ance. Este padrão, denominado **DOCSIS** (**D**ata **O**ver **C**able **S**ervice **I**nterface **S**pecificaçāo), substituiu principalmente os modems proprietários. DOCSIS versão 1.0 veio lançado em 1997, e logo foi seguido pelo DOCSIS 2.0 em 2001. Ele aumentou taxas de fluxo para melhor oferecer suporte a serviços simétricos, como telefonia IP. A maioria versão recente do padrão é o DOCSIS 3.0, que saiu em 2006. Ele usa mais largura de banda para aumentar as taxas em ambas as direções. A versão europeia de esses padrões são chamados de **EuroDOCSIS** . Nem todas as operadoras de cabo gostam da ideia de um padrão, no entanto, uma vez que muitos deles estavam ganhando um bom dinheiro com o leasing de seus modems para seus clientes cativos. Um padrão aberto com dezenas de fabricantes A venda de modems a cabo em lojas acaba com essa prática lucrativa. A interface do modem para o computador é direta. Normalmente é Ethernet, ou ocasionalmente USB. A outra extremidade é mais complicada, pois usa todo o FDM, TDM e CDMA para compartilhar a largura de banda do cabo entre os assinantes. Quando um modem a cabo é conectado e ligado, ele verifica o downstream canais à procura de um pacote especial enviado periodicamente pelo headend para provide parâmetros do sistema para modems que acabaram de ficar online. Ao encontrar este pacote, o novo modem anuncia sua presença em um dos canais upstream. O headend responde atribuindo o modem ao seu upstream e downstream canais. Essas atribuições podem ser alteradas posteriormente, se o headend considerar necessário necessário para equilibrar a carga. O uso de canais de 6 MHz ou 8 MHz é parte do FDM. Cada modem a cabo envia dados em um canal upstream e um canal downstream, ou canais múltiplos

Página 208

184

A CAMADA FÍSICA INDIVÍDUO. 2

sob DOCSIS 3.0. O esquema usual é levar cada 6 (ou 8) MHz downstream canal e module-o com QAM-64 ou, se a qualidade do cabo for excepcionalmente bom, QAM-256. Com um canal de 6 MHz e QAM-64, obtemos cerca de 36 Mbps. Quando a sobrecarga é subtraída, a carga útil líquida é de cerca de 27 Mbps. Com QAM-256, a carga útil líquida é de cerca de 39 Mbps. Os valores europeus são 1/3 maiores. Para upstream, há mais ruído de RF porque o sistema não era originalmente projetado para dados, e o ruído de vários assinantes é canalizado para o headend, portanto, um esquema mais conservador é usado. Isso varia de QPSK a QAM-128, onde alguns dos símbolos são usados para proteção de erro com Trellis Coded Modulation. Com menos bits por símbolo no upstream, a assimetria entre As taxas a montante e a jusante são muito mais do que o sugerido pela Figura 2.52. O TDM é então usado para compartilhar largura de banda no upstream em vários sub esribas. Caso contrário, suas transmissões colidiriam no headend. O tempo é fornecidos em **minijanelas** e diferentes assinantes enviam em minislots diferentes. Para fazer isso funcionar, o modem determina sua distância do headend enviando é um pacote especial e vê quanto tempo leva para obter a resposta. Este processo é chamado de **alcance** . É importante para o modem saber sua distância para obter o tempo certo. Cada pacote upstream deve caber em um ou mais minislots consecutivos no headend quando for recebido. O headend anuncia o início de um novo rodada de minislots periodicamente, mas o tiro de partida não é ouvido em todos os modems simultaneamente devido ao tempo de propagação ao longo do cabo. Sabendo quão longe é do headend, cada modem pode calcular há quanto tempo o primeiro minislot realmente começou. O comprimento do minislot depende da rede. Uma carga útil típica é de 8 bytes. Durante a inicialização, o headend atribui cada modem a um minislot para usar para solicitando largura de banda upstream. Quando um computador deseja enviar um pacote, ele transfere o pacote para o modem, que então solicita o número necessário de minislots para isso. Se a solicitação for aceita, o headend coloca uma confirmação no canal downstream informando ao modem quais minislots foram reservados

para seu pacote. O pacote é então enviado, começando no minislot alocado a ele. De Anúncios-pacotes adicionais podem ser solicitados usando um campo no cabeçalho.

Como regra, vários modems serão atribuídos ao mesmo minislot, o que leva a contenção. Existem duas possibilidades diferentes para lidar com isso. O primeiro é aquele CDMA é usado para compartilhar o minislot entre assinantes. Isso resolve o problema problema de atenção porque todos os assinantes com uma sequência de código CDMA podem enviar em

ao mesmo tempo, embora a taxa reduzida. A segunda opção é que o CDMA não é usado, caso em que pode não haver reconhecimento da solicitação devido a um colisão. Nesse caso, o modem apenas espera um tempo aleatório e tenta novamente. Depois de a cada falha sucessiva, o tempo de randomização é dobrado. (Para leitores já um tanto familiarizado com redes, este algoritmo é apenas encaixado ALOHA com bi-backoff exponencial nary. Ethernet não pode ser usada no cabo porque as estações podem não sentir o meio. Voltaremos a essas questões no cap. 4.)

Os canais downstream são gerenciados de forma diferente do canal upstream nels. Para começar, há apenas um remetente (o headend), então não há contenção

Página 209

SEC. 2,8
TELEVISÃO À CABO

185

e sem necessidade de minislots, que na verdade são apenas multi-divisão de tempo estatísticoplexing. Por outro lado, a quantidade de tráfego downstream é geralmente muito maior do que o upstream, então um tamanho de pacote fixo de 204 bytes é usado. Parte disso é um Reed-Código de correção de erros Solomon e alguma outra sobrecarga, deixando uma carga útil do usuário de 184 bytes. Esses números foram escolhidos para compatibilidade com a televisão digital usando MPEG-2, para que os canais de TV e dados downstream sejam formatados da mesma forma maneira. Logicamente, as conexões são conforme ilustradas na Figura 2-53.

Figura 2-53. Detalhes típicos dos canais a montante e a jusante no Norte América.

2.8.5 ADSL Versus Cabo

Qual é melhor, ADSL ou cabo? Isso é como perguntar qual sistema operacional é melhor. Ou qual idioma é melhor. Ou qual religião. Qual resposta você recebe depende de quem você pergunta. Vamos comparar ADSL e cabo em alguns pontos. Ambos usam fibra no backbone, mas diferem na borda. O cabo usa coaxial; ADSL usa par trançado. A capacidade de carga teórica do coaxial é de centenas de vezes mais do que par trançado. No entanto, a capacidade total do cabo não está disponível capaz para usuários de dados porque grande parte da largura de banda do cabo é desperdiçada em coisas como programas de televisão.

Na prática, é difícil generalizar sobre a capacidade efetiva. Provedores ADSL dar declarações específicas sobre a largura de banda (por exemplo, 1 Mbps downstream, 256 kbps a montante) e geralmente atingem cerca de 80% dele de forma consistente. Provedores de cabo pode limitar artificialmente a largura de banda de cada usuário para ajudá-los a melhorar o desempenho

previsões, mas eles não podem realmente dar garantias porque a capacidade efetiva depende de quantas pessoas estão atualmente ativas no segmento de cabo do usuário. Às vezes, pode ser melhor do que ADSL e às vezes pode ser pior. o que pode ser irritante, porém, é a imprevisibilidade. Tendo um ótimo serviço um minuto não garante um ótimo serviço no minuto seguinte, pois o maior consumo de largura de banda na cidade pode ter acabado de ligar o computador.

Página 210

186
A CAMADA FÍSICA
INDIVÍDUO. 2

À medida que um sistema ADSL adquire mais usuários, seu número crescente tem pouco efeito sobre os usuários existentes, uma vez que cada usuário tem uma conexão dedicada. Com cabo,

à medida que mais assinantes se inscrevem no serviço de Internet, desempenho para os usuários existentes

Vai cair. A única solução é o operador de cabo dividir os cabos ocupados e conectar cada um para um nó de fibra diretamente. Fazer isso custa tempo e dinheiro, então há pressões de negócios para evitá-lo.

À parte, já estudamos outro sistema com um canal compartilhado

como o cabo: o sistema de telefonia móvel. Aqui também, um grupo de usuários - poderíamos chame-los de companheiros de cela - compartilhe uma quantidade fixa de largura de banda. Para tráfego de voz, que

é bastante suave, a largura de banda é rigidamente dividida em blocos fixos entre os ativos usuários usando FDM e TDM. Mas para o tráfego de dados, essa divisão rígida é muito ineficaz suficiente porque os usuários de dados estão frequentemente ociosos, caso em que sua banda reservada

largura é desperdiçada. Tal como acontece com o cabo, um meio mais dinâmico é usado para alocar o

largura de banda compartilhada.

A disponibilidade é um problema no qual ADSL e cabo diferem. Todo mundo tem um telefone, mas nem todos os usuários estão próximos o suficiente de suas estações finais para obter ADSL. No

por outro lado, nem todo mundo tem cabo, mas se você tiver cabo e a empresa provides acesso à Internet, você pode obtê-lo. A distância para o nó de fibra ou headend não é um problema. Também é importante notar que, como o cabo começou como uma distribuidora de televisão

meio de ação, poucas empresas o têm.

Por ser um meio ponto a ponto, o ADSL é inherentemente mais seguro do que o cabo.

Qualquer usuário de cabo pode ler facilmente todos os pacotes que passam pelo cabo. Por esta razão filho, qualquer provedor de cabo decente criptografará todo o tráfego em ambas as direções. Nunca-no entanto, fazer com que seu vizinho receba suas mensagens criptografadas ainda é menos seguro do que

tê-lo sem receber nada.

O sistema telefônico é geralmente mais confiável do que o cabo. Por exemplo, tem energia de reserva e continua a funcionar normalmente mesmo durante uma queda de energia. Com o cabo, se a energia de qualquer amplificador ao longo da cadeia falhar, tudo a jusante os usuários são desligados instantaneamente.

Finalmente, a maioria dos provedores de ADSL oferece uma escolha de ISPs. Às vezes são mesmo obrigado a fazê-lo por lei. Nem sempre é o caso com operadoras de cabo.

A conclusão é que ADSL e cabo são muito mais semelhantes do que diferentes. Ferent. Eles oferecem um serviço comparável e, como a competição entre eles aquece preços provavelmente comparáveis.

2.9 RESUMO

A camada física é a base de todas as redes. A natureza impõe dois fundamentos limites mentais em todos os canais, e estes determinam sua largura de banda. Esses limites são o limite de Nyquist, que lida com canais silenciosos, e o limite de Shannon, que lida com canais barulhentos.

Página 211

SEC. 2,9
RESUMO

187

Os meios de transmissão podem ser guiados ou não. A principal mídia guiada são par trançado, cabo coaxial e fibra óptica. A mídia não guiada inclui terres-rádio de teste, micro-ondas, infravermelho, lasers pelo ar e satélites.

Métodos de modulação digital enviam bits por meio de mídia guiada e não guiada como um sinais de log. Os códigos de linha operam em banda base e os sinais podem ser colocados em um banda de passagem modulando a amplitude, frequência e fase de uma portadora. Chann-Nels podem ser compartilhados entre usuários com tempo, frequência e divisão de código multiplexing.

Um elemento chave na maioria das redes de longa distância é o sistema telefônico. Seu principal os componentes são os loops, troncos e switches locais. ADSL oferece velocidades de até 40 Mbps sobre o loop local, dividindo-o em muitas subportadoras que funcionam em paralelo. Isso excede em muito as taxas de modems de telefone. PONs trazem fibra para o casa para taxas de acesso ainda maiores do que ADSL.

Os troncos carregam informações digitais. Eles são multiplexados com WDM para fornecer Sion muitos links de alta capacidade sobre fibras individuais, bem como com TDM para compartilhe cada link de alta taxa entre os usuários. Comutação de circuitos e pacote mudar são importantes.

Para aplicações móveis, o sistema de telefonia fixa não é adequado. Móvel os telefones são atualmente amplamente utilizados para voz e cada vez mais para dados. Eles passaram por três gerações. A primeira geração, 1G, era analógica e dominado por AMPS. 2G era digital, com o GSM atualmente o mais amplamente de-sistema de telefonia móvel utilizado no mundo. 3G é digital e baseado em banda larga CDMA, com WCDMA e também CDMA2000 em implantação.

Um sistema alternativo de acesso à rede é o sistema de televisão a cabo. Tem evoluiu gradualmente de cabo coaxial para cabo coaxial de fibra híbrido e de televisão para televisão e Internet. Potencialmente, ele oferece uma largura de banda muito alta, mas a banda largura na prática depende muito dos outros usuários porque é compartilhada.

PROBLEMAS

1. Calcule os coeficientes de Fourier para a função $f(t) = t$ ($0 \leq t \leq 1$).
2. Um canal silencioso de 4 kHz é amostrado a cada 1 ms. Qual é a taxa máxima de dados? Como a taxa de dados máxima muda se o canal for barulhento, com sinal-ruído proporção de 30 dB?
3. Os canais de televisão têm 6 MHz de largura. Quantos bits / s podem ser enviados se a digitação de quatro níveis sinais itálicos são usados? Assuma um canal silencioso.
4. Se um sinal binário for enviado por um canal de 3 kHz, cuja relação sinal-ruído é de 20 dB, qual é a taxa de dados máxima alcançável?
5. Qual relação sinal-ruído é necessária para colocar uma portadora T1 em uma linha de 50 kHz?
6. Quais são as vantagens da fibra óptica sobre o cobre como meio de transmissão? É Existe alguma desvantagem em usar fibra óptica em vez de cobre?

Página 212

188

A CAMADA FÍSICA INDIVÍDUO. 2

7. Quanta largura de banda existe em 0,1 micron de espectro em um comprimento de onda de 1 micron?
8. Deseja-se enviar uma sequência de imagens da tela do computador por meio de uma fibra óptica. o a tela é de 2560×1600 pixels, cada pixel tendo 24 bits. Existem 60 imagens de tela por segundo. Quanta largura de banda é necessária e quantos microns de comprimento de onda são necessário para esta banda em 1,30 microns?
9. O teorema de Nyquist é verdadeiro para fibra óptica monomodo de alta qualidade ou apenas para fio de cobre?
10. Antenas de rádio geralmente funcionam melhor quando o diâmetro da antena é igual ao da onda comprimento da onda de rádio. Antenas razoáveis variam de 1 cm a 5 metros de diâmetro ter. Que faixa de frequência isso cobre?
11. Um feixe de laser de 1 mm de largura é direcionado a um detector de 1 mm de largura a 100 m de distância no telhado de um prédio. Quanto de desvio angular (em graus) o laser deve ter antes de perder o detector?
12. Os 66 satélites de órbita baixa no projeto Iridium são divididos em seis colares ao redor a Terra. Na altitude que estão usando, o período é de 90 minutos. Qual é a média intervalo para handoffs para um transmissor estacionário?
13. Calcule o tempo de trânsito de ponta a ponta para um pacote para ambos GEO (altitude: 35.800 km), Satélites MEO (altitude: 18.000 km) e LEO (altitude: 750 km).
14. Qual é a latência de uma chamada originada no Pólo Norte para chegar ao Pólo Sul se a chamada é encaminhada através de satélites Iridium? Suponha que o tempo de comutação nos satélites seja 10 microsegundos e o raio da terra é 6.371 km.
15. Qual é a largura de banda mínima necessária para atingir uma taxa de dados de B bits / s se o sinal é transmitido usando a codificação NRZ, MLT-3 e Manchester? Explique o seu responda.

- 16.** Prove que na codificação 4B / 5B, uma transição de sinal ocorrerá pelo menos a cada quatro bits vezes.
- 17.** Quantos códigos de estação final existiam antes de 1984, quando cada estação final era nomeada por seu código de área de três dígitos e os três primeiros dígitos do número local? Códigos de área começava com um dígito no intervalo de 2 a 9, tinha 0 ou 1 como o segundo dígito e terminava com qualquer dígito. Os primeiros dois dígitos de um número local sempre estiveram no intervalo 2-9. o o terceiro dígito pode ser qualquer dígito.
- 18.** Um sistema telefônico simples consiste em duas estações finais e uma única estação interurbana para a qual cada estação final é conectada por um tronco full-duplex de 1 MHz. O telefone médio é costumava fazer quatro chamadas por dia de trabalho de 8 horas. A duração média da chamada é de 6 minutos. Dez por cento das chamadas são de longa distância (ou seja, passam pela central de pedágio). O que é número máximo de telefones que uma estação final pode suportar? (Suponha 4 kHz por circuito.) Explique por que uma companhia telefônica pode decidir oferecer suporte a um número menor de telefones do que este número máximo na estação final.
- 19.** Uma companhia telefônica regional possui 10 milhões de assinantes. Cada um de seus telefones é conectado a um escritório central por um par trançado de cobre. O comprimento médio destes pares trançados é de 10 km. Quanto vale o cobre nos loops locais? Presumir

Página 213

INDIVÍDUO. 2 PROBLEMAS

189

- que a seção transversal de cada fio é um círculo de 1 mm de diâmetro, a densidade do cobre é 9,0 gramas / cm³, e esse cobre é vendido por US \$ 6 o quilo.
- 20.** É um oleoduto um sistema simplex, um sistema half-duplex, um sistema full-duplex ou nenhuma das acima? Que tal um rio ou uma comunicação estilo walkie-talkie?
- 21.** O custo de um microprocessador rápido caiu a um ponto em que agora é possível coloque um em cada modem. Como isso afeta o tratamento de erros de linha telefônica? Isso nega a necessidade de verificação / correção de erros na camada 2?
- 22.** Um diagrama de constelação de modem semelhante à Fig. 2-23 tem pontos de dados nos seguintes coordenadas: (1, 1), (1, -1), (-1, 1) e (-1, -1). Quantos bps pode um modem com esses parâmetros alcançar 1200 símbolos / segundo?
- 23.** Qual é a taxa de bits máxima atingível em um modem padrão V.32 se a taxa de baud é 1200 e nenhuma correção de erro é usada?
- 24.** Quantas frequências um modem QAM-64 full-duplex usa?
- 25.** Dez sinais, cada um exigindo 4000 Hz, são multiplexados em um único canal usando FDM. Qual é a largura de banda mínima necessária para o canal multiplexado? Como suponha que as bandas de guarda tenham 400 Hz de largura.
- 26.** Por que o tempo de amostragem PCM foi definido em 125 usec?
- 27.** Qual é a sobrecarga percentual em uma portadora T1? Ou seja, qual porcentagem dos 1,544 Mbps não são entregues ao usuário final? Como isso se relaciona com a sobrecarga percentual em OC-1 ou linhas OC-768?
- 28.** Compare a taxa de dados máxima de um canal silencioso de 4 kHz usando
(a) Codificação analógica (por exemplo, QPSK) com 2 bits por amostra.
(b) O sistema T1 PCM.
- 29.** Se um sistema de portadora T1 escorregar e perder o controle de onde está, ele tenta resincronizar usando o primeiro bit em cada quadro. Quantos quadros terão que ser inspecionados em média para resincronizar com uma probabilidade de 0,001 de estar errado?
- 30.** Qual é a diferença, se houver, entre a parte demodulador de um modem e o codificador parte de um codec? (Afinal, ambos convertem sinais analógicos em digitais.)
- 31.** Os relógios SONET têm uma taxa de deriva de cerca de 1 parte em 10⁹. Quanto tempo leva para o deriva para igualar a largura de 1 bit? Você vê alguma implicação prática deste cálculo ção? Se sim, o quê?
- 32.** Quanto tempo leva para transmitir um arquivo de 1 GB de um VSAT para outro usando um hub conforme mostrado na Figura 2-17? Suponha que o uplink seja de 1 Mbps, o downlink seja de 7 Mbps, e a comutação de circuito é usada com tempo de configuração de circuito de 1,2 seg.
- 33.** Calcule o tempo de transmissão no problema anterior se a comutação de pacotes for usada. Suponha que o tamanho do pacote seja de 64 KB, o atraso de comutação no satélite e hub é de 10 microssegundos, e o tamanho do cabeçalho do pacote é de 32 bytes.
- 34.** Na Figura 2-40, a taxa de dados do usuário para OC-3 é indicada como 148,608 Mbps. Mostre como isso o número pode ser derivado dos parâmetros SONET OC-3. Qual será o bruto, SPE e taxas de dados do usuário de uma linha OC-3072?

Página 214

190

A CAMADA FÍSICA INDIVÍDUO. 2

35. Para acomodar taxas de dados mais baixas do que STS-1, a SONET tem um sistema de tributo virtual taries (VTs). Um VT é uma carga parcial que pode ser inserida em um quadro STS-1 e combinado com outras cargas parciais para preencher o quadro de dados. VT1.5 usa 3 colunas, VT2 usa 4 colunas, VT3 usa 6 colunas e VT6 usa 12 colunas de um STS-1 quadro, Armação. Qual VT pode acomodar

- (a) Um serviço DS-1 (1,544 Mbps)?
- (b) Serviço CEPT-1 europeu (2,048 Mbps)?
- (c) Um serviço DS-2 (6,312 Mbps)?

36. Qual é a largura de banda do usuário disponível em uma conexão OC-12c?

37. Três redes de comutação de pacotes contêm, cada uma, n nós. A primeira rede tem uma estrela topologia com um switch central, o segundo é um anel (bidirecional), e o terceiro é totalmente interconectado, com um fio de cada nó para todos os outros nós. O que são as caminhos de transmissão de melhor, média e pior caso em saltos?

38. Compare o atraso no envio de uma mensagem de x - bits em um caminho k -hop em um circuito comutado rede e em uma rede comutada por pacotes (levemente carregada). O tempo de configuração do circuito é s seg, o atraso de propagação é d seg por salto, o tamanho do pacote é p bits, e a taxa de dados é b bps. Em que condições a rede de pacotes tem um atraso menor? Além disso, simplesmente as condições sob as quais uma rede comutada por pacotes é preferível a um circuito rede comutada.

39. Suponha que x bits de dados do usuário sejam transmitidos por um caminho k -hop em um pacote-rede comutada como uma série de pacotes, cada um contendo p bits de dados e h bits de cabeçalho, com $x \gg p + h$. A taxa de bits das linhas é b bps e o atraso de propagação é negligenciável ble. Qual valor de p minimiza o atraso total?

40. Em um sistema de telefonia móvel típico com células hexagonais, é proibido reutilizar um fre-banda de frequência em uma célula adjacente. Se 840 frequências estão disponíveis, quantas podem ser usado em uma determinada célula?

41. O layout real das células raramente é tão regular quanto mostrado na Figura 2.45. Mesmo o formas de células individuais são tipicamente irregulares. Dê uma possível razão para isso pode ser. Como essas formas irregulares afetam a atribuição de frequência a cada célula?

42. Faça uma estimativa aproximada do número de microcélulas PCS de 100 m de diâmetro que seria levar para cobrir San Francisco (120 km quadrados).

43. Às vezes, quando um usuário móvel cruza a fronteira de uma célula para outra, o a chamada de aluguel é encerrada abruptamente, embora todos os transmissores e receptores estejam funcionando funcionamento perfeito. Por quê?

44. Suponha que A , B e C estejam transmitindo simultaneamente 0 bits, usando um sistema CDMA com as sequências de chips da Figura 2.28 (a). Qual é a sequência de chips resultante?

45. Considere uma maneira diferente de olhar para a propriedade de ortogonalidade do chip CDMA sequências. Cada bit em um par de sequências pode corresponder ou não. Expresse o orto-propriedade de gonalidade em termos de correspondências e incompatibilidades.

46. Um receptor CDMA obtém os seguintes chips: (-1 +1 -3 +1 -1 -1 -3 +1 +1). Assumindo as sequências de chips definidas na Figura 2-28 (a), quais estações transmitiram e quais bits cada um enviou?

Página 215

INDIVÍDUO. 2

PROBLEMAS

191

47. Na Figura 2-28, há quatro estações que podem transmitir. Suponha mais quatro estações são adicionados. Fornece as sequências de chips dessas estações.

48. Na extremidade inferior, o sistema telefônico é em forma de estrela, com todos os loops locais em um vizinho borhood convergindo em uma estação final. Em contraste, a televisão a cabo consiste em um único um longo cabo serpenteando por todas as casas no mesmo bairro. Suponha que um futuro cabo de TV seria fibra de 10 Gbps em vez de cobre. Pode ser usado para simular o modelo de telefone em que todos têm sua própria linha privada para a estação final? E se então, quantas casas de um telefone poderiam ser conectadas a uma única fibra?

49. Uma empresa de TV a cabo decide fornecer acesso à Internet por cabo em um bairro construção de 5000 moradias. A empresa usa um cabo coaxial e alocação de espectro al-largura de banda downstream de 100 Mbps por cabo. Para atrair clientes, o empresa decide garantir pelo menos 2 Mbps de largura de banda downstream para cada casa em qualquer Tempo. Descreva o que a empresa de TV a cabo precisa fazer para fornecer essa garantia.

50. Usando a alocação espectral mostrada na Figura 2-52 e as informações fornecidas no texto, quantos Mbps um sistema de cabo aloca para upstream e quantos para down-

corrente?

51. Quão rápido um usuário de cabo pode receber dados se a rede estiver ociosa? Assuma isso a interface do usuário é

- (a) Ethernet de 10 Mbps
- (b) Ethernet de 100 Mbps
- (c) 54 Mbps sem fio.

52. Multiplexar múltiplos fluxos de dados STS-1, chamados tributários, desempenha um papel importante em SONET. Um multiplexador 3: 1 multiplexa três tributários STS-1 de entrada em um de saída colocar fluxo STS-3. Essa multiplexação é feita byte por byte. Ou seja, os três primeiros de put bytes são os primeiros bytes dos tributários 1, 2 e 3, respectivamente. os próximos três fora put bytes são os segundos bytes dos tributários 1, 2 e 3, respectivamente, e assim por diante. Escrever um programa que simula este multiplexador 3: 1. Seu programa deve consistir em cinco processos. O processo principal cria quatro processos, um para cada um dos três STS-1 tributários e um para o multiplexador. Cada processo tributário lê em um quadro STS-1 de um arquivo de entrada como uma sequência de 810 bytes. Eles enviam seus quadros (byte por byte) para o processo do multiplexador. O processo do multiplexador recebe esses bytes e gera uma Quadro STS-3 (byte por byte) gravando-o na saída padrão. Use tubos para comunicação cação entre processos.

53. Escreva um programa para implementar CDMA. Suponha que o comprimento de uma sequência de chips é oito e o número de estações transmitindo é quatro. Seu programa consiste em três conjuntos de processos: quatro processos de transmissor (t_0, t_1, t_2 e t_3), um processo de junção e quatro processos do receptor (r_0, r_1, r_2 e r_3). O programa principal, que também atua como o processo de junção primeiro lê quatro sequências de chips (notação bipolar) do padrão entrada e uma sequência de 4 bits (1 bit por processo do transmissor a ser transmitido), e bifurca quatro pares de processos de transmissor e receptor. Cada par de transmissor / receptor ceiver ($t_0, r_0; t_1, r_1; t_2, r_2; t_3, r_3$) são atribuídos a uma sequência de chips e cada o processo do transmissor é atribuído a 1 bit (primeiro bit para t_0 , segundo bit para t_1 e assim por diante). Próximo, cada processo do transmissor calcula o sinal a ser transmitido (uma sequência de 8 bits) e o envia para o processo de junção. Depois de receber sinais de todos os quatro transmissores processos, o processo de junção combina os sinais e envia o sinal combinado para

Página 216

192

A CAMADA FÍSICA INDIVÍDUO. 2

os quatro processos do receptor. Cada processo receptor então calcula o bit que tem recebido e imprime na saída padrão. Use tubos para comunicação entre processos.

Página 217

3

A CAMADA DE LINK DE DADOS

Neste capítulo, estudaremos os princípios de design para a segunda camada em nosso modelo, a camada de enlace de dados. Este estudo trata de algoritmos para alcançar re-comunicação confiável e eficiente de unidades inteiras de informação chamadas frames (rath-mais do que bits individuais, como na camada física) entre duas máquinas adjacentes.

Por adjacente, queremos dizer que as duas máquinas estão conectadas por uma comunicação canal que atua conceitualmente como um fio (por exemplo, um cabo coaxial, linha telefônica ou Canal sem fio). A propriedade essencial de um canal que o torna " semelhante a um fio " é que os bits são entregues exatamente na mesma ordem em que são enviados.

A princípio, você pode pensar que esse problema é tão trivial que não há nada para estudo - a máquina *A* apenas coloca os bits no arame, e a máquina *B* apenas os pega fora. Infelizmente, os canais de comunicação cometem erros ocasionalmente. Pele-além disso, eles têm apenas uma taxa de dados finita e há uma propagação diferente de zero

atraso entre a hora em que um bit é enviado e a hora em que é recebido. Essas limitações têm implicações importantes para a eficiência da transferência de dados. Os protocolos usado para comunicações deve levar todos esses fatores em consideração. Estes os protocolos são o assunto deste capítulo.

Após uma introdução aos principais problemas de design presentes na camada de enlace, nós iniciará nosso estudo de seus protocolos observando a natureza dos erros e como eles podem ser detectados e corrigidos. Em seguida, estudaremos uma série de cada vez mais protocolos complexos, cada um resolvendo cada vez mais os problemas presentes neste camada. Finalmente, concluiremos com alguns exemplos de protocolos de enlace de dados.

193

Página 218

194

A CAMADA DE LINK DE DADOS
INDIVÍDUO. 3

3.1 QUESTÕES DE PROJETO DA CAMADA DE LINK DE DADOS

A camada de enlace de dados usa os serviços da camada física para enviar e receber bits nos canais de comunicação. Ele tem várias funções, incluindo:

1. Fornecimento de uma interface de serviço bem definida para a camada de rede.
2. Lidar com erros de transmissão.
3. Regular o fluxo de dados para que os receptores lentos não sejam sobrecarregados por remetentes rápidos.

Para cumprir esses objetivos, a camada de enlace de dados pega os pacotes que obtém do camada de rede e os encapsula em **quadros** para transmissão. Cada quadro contém um cabeçalho de quadro, um campo de carga útil para conter o pacote e um quadro reboque, conforme ilustrado na Fig. 3-1. O gerenciamento de quadros é o cerne do que camada de enlace de dados sim. Nas seções a seguir, examinaremos todos os itens acima mencionou os problemas em detalhes.

Reboque
Cabeçalho
Campo de carga útil
Quadro, Armação
Máquina de envio
Pacote
Pacote
Máquina receptora
Reboque
Cabeçalho
Campo de carga útil

Figura 3-1. Relação entre pacotes e frames.

Embora este capítulo seja explicitamente sobre a camada de enlace de dados e seus protocolos, muitos dos princípios que estudaremos aqui, como controle de erros e controle de fluxo, são encontrados no transporte e também em outros protocolos. Isso ocorre porque a confiabilidade é um

objetivo geral, e é alcançado quando todas as camadas trabalham juntas. Na verdade, em muitos redes, essas funções são encontradas principalmente nas camadas superiores, com o link de dados camada fazendo o trabalho mínimo que é "bom o suficiente". No entanto, não importa onde eles são encontrados, os princípios são praticamente os mesmos. Eles costumam aparecer em suas formas mais simples e puras na camada de link de dados, tornando este um bom lugar para examinar-los em detalhes.

3.1.1 Serviços prestados à camada de rede

A função da camada de enlace de dados é fornecer serviços à camada de rede. O principal serviço é a transferência de dados da camada de rede na fonte machine para a camada de rede na máquina de destino. Na máquina de origem está

Página 219

SEC. 3,1
PROBLEMAS DE DESIGN DA CAMADA DE LINK DE DADOS

195

uma entidade, chame-a de processo, na camada de rede que entrega alguns bits aos dados

camada de enlace para transmissão ao destino. O trabalho da camada de enlace é transmitir os bits para a máquina de destino para que possam ser entregues à rede camada de trabalho lá, como mostrado na Fig. 3-2 (a). A transmissão real segue o caminho da Figura 3-2 (b), mas é mais fácil pensar em termos de duas camadas de enlace de dados processando se comunicando usando um protocolo de link de dados. Por este motivo, vamos implicar Use citadamente o modelo da Fig. 3-2 (a) ao longo deste capítulo.

```
4  
3  
2  
1  
4  
3  
2  
1  
4  
3  
2  
1  
Host 1  
Host 2  
Host 1  
Host 2  
Virtual  
caminho de dados  
Real  
caminho de dados  
(uma)  
(b)
```

Figura 3-2. (a) Comunicação virtual. (b) Comunicação real.

A camada de enlace de dados pode ser projetada para oferecer vários serviços. O serviço real os vícios oferecidos variam de protocolo para protocolo. Três possibilidades razoáveis os laços que iremos considerar são:

1. Serviço sem conexão não reconhecido.
2. Serviço sem conexão reconhecido.
3. Serviço orientado a conexão confirmado.

O serviço sem conexão não reconhecido consiste em ter a fonte machine enviar frames independentes para a máquina de destino sem ter o máquina de destino reconhecê-los. Ethernet é um bom exemplo de link de dados camada que fornece essa classe de serviço. Nenhuma conexão lógica é estabelecida beforehand ou lançado posteriormente. Se um quadro for perdido devido a ruído na linha, não

Página 220

196

A CAMADA DE LINK DE DADOS INDIVÍDUO. 3

é feita uma tentativa de detectar a perda ou recuperá-la na camada de enlace de dados. Isto classe de serviço é apropriada quando a taxa de erro é muito baixa, então a recuperação é deixada para as camadas superiores. Também é apropriado para tráfego em tempo real, como voz, no qual dados atrasados são piores do que dados ruins.

O próximo passo em termos de confiabilidade é o serviço sem conexão reconhecido vice. Quando este serviço é oferecido, ainda não há conexões lógicas usadas, mas cada quadro enviado é reconhecido individualmente. Desta forma, o remetente sabe se um quadro chegou corretamente ou foi perdido. Se não chegou dentro de um intervalo de tempo especificado, ele pode ser enviado novamente. Este serviço é útil em vez de confiável

canais, como sistemas sem fio. 802.11 (WiFi) é um bom exemplo desta classe de serviço.

Talvez valha a pena enfatizar que fornecer reconhecimentos nos dados A camada de link é apenas uma otimização, nunca um requisito. A camada de rede pode algumas maneiras de enviar um pacote e esperar que ele seja reconhecido por seu par no remoto máquina. Se a confirmação não ocorrer antes que o cronômetro expire, o

o remetente pode simplesmente enviar a mensagem inteira novamente. O problema com esta estratégia é

que pode ser ineficiente. Os links geralmente têm um comprimento máximo de quadro estrito impostas pelo hardware e atrasos de propagação conhecidos. A camada de rede faz não conhece esses parâmetros. Ele pode enviar um pacote grande que está dividido em, digamos, 10 quadros, dos quais 2 são perdidos em média. Então, demoraria muito tempo para o pacote passar. Em vez disso, se quadros individuais são reconhecidos e retransmitidos, os erros podem ser corrigidos de forma mais direta e rápida. Em canais confiáveis, como fibra, a sobrecarga de um protocolo de link de dados pesado pode ser desnecessário, mas em canais sem fio (inerentemente não confiáveis) é bom vale o custo.

Voltando aos nossos serviços, o serviço mais sofisticado, a camada de enlace de dados pode fornecer à camada de rede é um serviço orientado à conexão. Com este serviço, as máquinas de origem e destino estabelecem uma conexão antes que quaisquer dados sejam transferido. Cada quadro enviado pela conexão é numerado, e o link de dados camada garante que cada quadro enviado seja realmente recebido. Além disso, garante mostra que cada quadro é recebido exatamente uma vez e que todos os quadros são recebidos em a ordem certa. O serviço orientado a conexão, portanto, fornece o processo de camada de rede esses com o equivalente a um fluxo de bits confiável. É apropriado por muito tempo, irrelacionados responsáveis, como um canal de satélite ou um circuito telefônico de longa distância. E se serviço sem conexão reconhecido foi usado, é concebível que o acesso perdido conhecimentos podem fazer com que um quadro seja enviado e recebido várias vezes, largura de banda.

Quando o serviço orientado à conexão é usado, as transferências passam por três diferentes fases. Na primeira fase, a conexão é estabelecida tendo ambos os lados inicializar variáveis e contadores necessários para manter o controle de quais quadros foram refeitos ceifadas e quais não. Na segunda fase, um ou mais frames são acionados finalmente transmitido. Na terceira e última fase, a conexão é liberada, liberando as variáveis, buffers e outros recursos usados para manter a conexão.

Página 221

SEC. 3,1

PROBLEMAS DE DESIGN DA CAMADA DE LINK DE DADOS

197

3.1.2 Enquadramento

Para fornecer serviço à camada de rede, a camada de enlace de dados deve usar o serviço vício fornecido a ele pela camada física. O que a camada física faz é aceitar um fluxo de bits brutos e tentar entregá-lo ao destino. Se o canal estiver barulhento, como é para a maioria dos links sem fio e alguns com fio, a camada física adicionará alguns redundância para seus sinais para reduzir a taxa de erro de bits a um nível tolerável. Contudo, o fluxo de bits recebido pela camada de enlace de dados não tem garantia de estar livre de erros. Alguns bits podem ter valores diferentes e o número de bits recebidos pode ser menor que, igual ou maior que o número de bits transmitidos. Depende dos dados camada de link para detectar e, se necessário, corrigir erros.

A abordagem usual é a camada de enlace de dados dividir o fluxo de bits em quadros discretos, calcule um token curto chamado checksum para cada quadro, e clua a soma de verificação no quadro quando ele é transmitido. (Algoritmos de checksum serão discutidos mais adiante neste capítulo.) Quando um quadro chega ao destino, a soma de verificação é recalculada. Se a soma de verificação recém-calculada for diferente de aquele contido no quadro, a camada de enlace de dados sabe que ocorreu um erro corrigido e toma medidas para lidar com isso (por exemplo, descartando o quadro ruim e possivelmente

também enviando de volta um relatório de erro).

Dividir o fluxo de bits em quadros é mais difícil do que parece à primeira vista.

Um bom design deve tornar mais fácil para um receptor encontrar o início de novos quadros enquanto usa pouca largura de banda do canal. Veremos quatro métodos:

1. Contagem de bytes.

2. Sinalizar bytes com preenchimento de bytes.

3. Sinalizar bits com enchimento de bits.

4. Violações de codificação da camada física.

O primeiro método de enquadramento usa um campo no cabeçalho para especificar o número de bytes no quadro. Quando a camada de enlace de dados no destino vê a contagem de bytes, ele sabe quantos bytes se seguem e, portanto, onde está o final do quadro. Isto técnica é mostrada na Fig. 3-3 (a) para quatro pequenos exemplos de quadros de tamanhos 5, 5, 8, e 8 bytes, respectivamente.

O problema com este algoritmo é que a contagem pode ser distorcida por uma transmissão erro de sessão. Por exemplo, se a contagem de bytes de 5 no segundo quadro da Fig. 3-3 (b) torna-se 7 devido a uma única inversão de bit, o destino sairá da sincronização ção. Assim, não será possível localizar o início correto do próximo quadro. Mesmo se o a soma de verificação está incorreta, então o destino sabe que o quadro está ruim, ele ainda não tem maneira de dizer onde começa o próximo quadro. Enviando um quadro de volta para a fonte pedir uma retransmissão também não ajuda, já que o destino não saiba quantos bytes ignorar para chegar ao início da retransmissão. Para por isso, o método de contagem de bytes raramente é usado sozinho.

Página 222

198

A CAMADA DE LINK DE DADOS

INDIVÍDUO. 3

(b)

(uma)

5

1 2 3 4 5 6 7 8 9 8 0 1 2 3 4 5 6 8 7 8 9 0 1 2 3

5 1 2 3 4 7 6 7 8 9 8 0 1 2 3 4 5 6 8 7 8 9 0 1 2 3

Contagem de bytes

Um byte

Erro

Quadro 1

5 bytes

Quadro 1

Quadro 2

5 bytes

Quadro 2

(Errado)

Quadro 3

8 bytes

Quadro 4

8 bytes

Agora um byte

contagem

Figura 3-3. Um fluxo de bytes. (a) Sem erros. (b) Com um erro.

O segundo método de enquadramento contorna o problema de ressincronização após um erro, fazendo com que cada quadro inicie e termine com bytes especiais. Frequentemente o mesmo byte, denominado **byte de sinalização**, é usado como delimitador inicial e final. Esse byte é mostrado na Fig. 3-4 (a) como FLAG. Dois bytes de flag consecutivos indicam o fim de um quadro e o início do seguinte. Assim, se o receptor perder sincronização, ele pode apenas procurar por dois bytes de flag para encontrar o fim do atual quadro e o início do próximo quadro.

Porém, ainda há um problema que temos que resolver. Pode acontecer que o sinalizador byte ocorre nos dados, especialmente quando dados binários, como fotografias ou canções estão sendo transmitidas. Essa situação interferiria no enquadramento.¹ maneira de resolver este problema é fazer com que a camada de enlace do remetente insira um escape byte (ESC) imediatamente antes de cada byte de sinalizador "acidental" nos dados. Assim, um

O byte da bandeira de enquadramento pode ser distinguido de um nos dados pela ausência ou presença de um byte de escape antes dele. A camada de enlace de dados na extremidade receptora re-

move os bytes de escape antes de fornecer os dados à camada de rede. Esta técnica que é chamado de **enchimento de bytes**.

Claro, a próxima pergunta é: o que acontece se um byte de escape ocorrer no meio dos dados? A resposta é que ele também é preenchido com um byte de escape. Em

o receptor, o primeiro byte de escape é removido, deixando o byte de dados que o segue (que pode ser outro byte de escape ou o byte de sinalização). Alguns exemplos são mostrados na Fig. 3-4 (b). Em todos os casos, a sequência de bytes entregue após destuffing é exatamente o mesmo que a sequência de bytes original. Ainda podemos pesquisar um limite de quadro procurando por dois bytes de flag em uma linha, sem se preocupar em desfazer escapes. O esquema de preenchimento de bytes representado na Fig. 3-4 é uma ligeira simplificação do usado em PPP (**protocolo ponto a ponto**), que é usado para transportar pacotes links de comunicação. Discutiremos o PPP próximo ao final deste capítulo.

Página 223

SEC. 3,1

PROBLEMAS DE DESIGN DA CAMADA DE LINK DE DADOS

199

```

UMA
B
ESC
BANDEIRA
UMA
B
ESC
ESC
UMA
ESC
B
ESC
ESC
BANDEIRA
UMA
ESC
B
ESC
ESC
B
UMA
BANDEIRA
B
UMA
ESC
BANDEIRA
UMA
ESC
B
ESC
UMA
ESC
B
BANDEIRA
Reboque
FLAG Header
Campo de carga útil
Bytes originais
Depois de encher
(uma)
(b)

```

Figura 3-4. (a) Um quadro delimitado por bytes de sinalizador. (b) Quatro exemplos de byte seqüências antes e depois do preenchimento de bytes.

O terceiro método de delimitar o fluxo de bits contorna a desvantagem de preenchimento de bytes, que é vinculado ao uso de bytes de 8 bits. O enquadramento também pode ser

ser feito no nível de bits, para que os quadros possam conter um número arbitrário de bits compostos de unidades de qualquer tamanho. Foi desenvolvido para o **HDLC** (**High-protocol Data Link Control**). Cada quadro começa e termina com um especial padrão de bits, 01111110 ou 0x7E em hexadecimal. Este padrão é um byte de sinalizador. Quando-sempre que a camada de enlace de dados do remetente encontra cinco 1s consecutivos nos dados, insere automaticamente um bit 0 no fluxo de bits de saída. Este **recheio** é analógous to byte stuffing, em que um byte de escape é inserido no caractere de saída ter stream antes de um byte de flag nos dados. Também garante uma densidade mínima de transições que ajudam a camada física a manter a sincronização. USB (universal Serial Bus) usa bit stuffing por esse motivo.

Quando o receptor vê cinco bits 1 de entrada consecutivos, seguidos por um bit 0,

ele remove automaticamente (isto é, apaga) o bit 0. Assim como o enchimento de bytes é completamente transparente para a camada de rede em ambos os computadores, portanto, é o enchimento de bits. Se o usuário os dados contêm o padrão de sinalizador, 01111110, este sinalizador é transmitido como 011111010, mas armazenado na memória do receptor como 01111110. A Figura 3-5 dá um exemplo de bit estofamento. Com o bit stuffing, o limite entre dois quadros pode ser inequivocamente reconhecido pelo padrão da bandeira. Assim, se o receptor perder a noção de onde está, todos tem que fazer é varrer a entrada para sequências de flag, uma vez que elas só podem ocorrer no quadro limites e nunca dentro dos dados.

Página 224

200

A CAMADA DE LINK DE DADOS

INDIVÍDUO. 3

01101111111111111111110010
0110111101111011111010010

Pedaços recheados
(uma)

(b)

(c) 01101111111111111111110010

Figura 3-5. Recheio de pouco. (a) Os dados originais. (b) Os dados conforme aparecem em a linha. (c) Os dados como eles são armazenados na memória do receptor após a destruição fing.

Com o preenchimento de bits e bytes, um efeito colateral é que o comprimento de um quadro agora depende do conteúdo dos dados que carrega. Por exemplo, se não houver bandeira bytes nos dados, 100 bytes podem ser transportados em um quadro de aproximadamente 100 bytes. E se, no entanto, os dados consistem apenas em bytes de sinalização, cada byte de sinalização terá escape e

o quadro terá aproximadamente 200 bytes de comprimento. Com o recheio de bits, o aumento seria aproximadamente 12,5%, pois 1 bit é adicionado a cada byte.

O último método de enquadramento é usar um atalho da camada física. Nós viu no cap. 2 que a codificação de bits como sinais geralmente inclui redundância para ajude o receptor. Esta redundância significa que alguns sinais não ocorrerão no registro dados gerais. Por exemplo, no código de linha 4B / 5B 4 bits de dados são mapeados para 5 sinais bits para garantir transições de bits suficientes. Isso significa que 16 dos 32 sinais possibilidades não são usadas. Podemos usar alguns sinais reservados para indicar o início e fim dos frames. Na verdade, estamos usando "violações de codificação" para delimitar os quadros.

A beleza desse esquema é que, por serem sinais reservados, é fácil encontrar o início e o fim dos quadros e não há necessidade de preencher os dados.

Muitos protocolos de enlace de dados usam uma combinação desses métodos para segurança. UMA padrão comum usado para Ethernet e 802.11 é ter um quadro começando com um padrão bem definido denominado **préambulo**. Este padrão pode ser bastante longo (72 bits é típico para 802.11) para permitir que o receptor se prepare para um pacote de entrada. o préambulo é seguido por um campo de comprimento (ou seja, contagem) no cabeçalho que é usado para localizar o final do quadro.

3.1.3 Controle de Erro

Tendo resolvido o problema de marcar o início e o fim de cada quadro, nós chegamos ao próximo problema: como garantir que todos os quadros sejam eventualmente entregues ao a camada de rede no destino e na ordem adequada. Suponha para o momento em que o receptor pode dizer se um quadro que recebe contém ou informações defeituosas (veremos os códigos usados para detectar e corrigir erros de transmissão na seção 3.2). Para serviço sem conexão não reconhecido,

pode ser bom se o remetente apenas continuar enviando frames sem levar em conta se

Página 225

SEC. 3,1

PROBLEMAS DE DESIGN DA CAMADA DE LINK DE DADOS

201

eles estavam chegando corretamente. Mas para um serviço orientado a conexão confiável, não estar bem de jeito nenhum.

A maneira usual de garantir uma entrega confiável é fornecer ao remetente alguns feedback sobre o que está acontecendo do outro lado da linha. Normalmente, o protocolo exige que o receptor envie de volta quadros de controle especiais com positividade ou confirmações negativas sobre os quadros recebidos. Se o remetente receber um reconhecimento positivo sobre um quadro, ele sabe que o quadro chegou com segurança. Por outro lado, um reconhecimento negativo significa que algo foi errado e o quadro deve ser transmitido novamente.

Uma complicação adicional vem da possibilidade de problemas de hardware que podem fazer com que um quadro desapareça completamente (por exemplo, em uma explosão de ruído). Neste caso, o

receptor não reagirá de forma alguma, uma vez que não há razão para reagir. Da mesma forma, se o acquaintance quadro de conhecimento é perdido, o remetente não saberá como proceder. Deveria deixar claro que um protocolo em que o remetente transmite um quadro e depois espera por um reconhecimento, positivo ou negativo, ficará suspenso para sempre se um quadro for perdido devido a, por exemplo, hardware com defeito ou canal de comunicação com defeito.

Esta possibilidade é tratada introduzindo temporizadores na camada de enlace de dados.

Quando o remetente transmite um quadro, geralmente também inicia um cronômetro. O cronômetro é

definido para expirar após um intervalo longo o suficiente para o quadro chegar ao destino, ser processado lá, e ter a confirmação propagada de volta para o remetente.

Normalmente, o quadro será recebido corretamente e a confirmação será antes que o cronômetro se esgote, caso em que o cronômetro será cancelado.

No entanto, se o quadro ou a confirmação forem perdidos, o cronômetro irá desligado, alertando o remetente sobre um problema potencial. A solução óbvia é apenas transmitir o quadro novamente. No entanto, quando os quadros podem ser transmitidos, várias vezes há o perigo de o receptor aceitar o mesmo quadro dois ou mais vezes e passá-lo para a camada de rede mais de uma vez. Para evitar que isso aconteça, geralmente é necessário atribuir números de sequência aos quadros de saída, para que o receptor possa distinguir as retransmissões dos originais.

Toda a questão de gerenciar os temporizadores e números de sequência, de modo a garantir que cada quadro é finalmente passado para a camada de rede no destino exatamente uma vez, nem mais nem menos, é uma parte importante das funções da camada de enlace de dados (e camadas superiores). Mais adiante neste capítulo, veremos uma série de exemplos sofisticados para ver como esse gerenciamento é feito.

3.1.4 Controle de fluxo

Outro problema de design importante que ocorre na camada de enlace de dados (e superior camadas também) é o que fazer com um remetente que deseja transmitir sistematicamente quadros mais rápido do que o receptor pode aceitá-los. Esta situação pode ocorrer quando o remetente está sendo executado em um computador rápido e poderoso e o receptor está sendo executado em uma máquina lenta e de baixo custo. Uma situação comum é quando um smartphone solicita uma Página da Web de um servidor muito mais poderoso, que então liga a mangueira de incêndio e

Página 226

202

A CAMADA DE LINK DE DADOS

INDIVÍDUO. 3

envia os dados para o pobre telefone indefeso até que ele esteja completamente inundado. Mesmo se a transmissão está livre de erros, o receptor pode ser incapaz de lidar com os quadros como rápido quanto eles chegam e vão perder alguns.

Obviamente, algo deve ser feito para evitar essa situação. Duas abordagens são comumente usados. No primeiro, **controle de fluxo baseado em feedback**, o receptor envia de volta informações ao remetente, dando-lhe permissão para enviar mais dados, ou em menos dizendo ao remetente como está o receptor. No segundo, **com base em taxas controle de fluxo**, o protocolo tem um mecanismo embutido que limita a taxa na qual os remetentes podem transmitir dados, sem usar feedback do receptor.

Neste capítulo, estudaremos esquemas de controle de fluxo baseados em feedback, principalmente porque os esquemas baseados em taxas são vistos apenas como parte da camada de transporte (Cap. 5).

Esquemas baseados em feedback são vistos tanto na camada de link quanto nas camadas superiores. o

último é mais comum nos dias de hoje, caso em que o hardware da camada de link é desenhado para funcionar rápido o suficiente para não causar perda. Por exemplo, hardware implementações da camada de link como **NICs (Network Interface Cards)** são algumas tempos ditos para rodar na " velocidade do fio ", o que significa que eles podem lidar com frames tão rápido quanto

eles podem chegar no link. Qualquer overruns não é um problema de link, então eles são tratado por camadas superiores.

Vários esquemas de controle de fluxo baseados em feedback são conhecidos, mas a maioria deles usar o mesmo princípio básico. O protocolo contém regras bem definidas sobre quando um remetente pode transmitir o próximo quadro. Essas regras geralmente proíbem frames de ser enviado até que o destinatário conceda permissão, implícita ou explicitamente. Por exemplo, quando uma conexão é configurada, o receptor pode dizer: " Você pode me enviar n frames agora, mas depois que eles forem enviados, não envie mais até que eu lhe diga para continuar. " Examinaremos os detalhes em breve.

3.2 DETECCÃO E CORREÇÃO DE ERROS

Vimos no cap. 2 que os canais de comunicação têm uma variedade de características. Alguns canais, como fibra óptica em redes de telecomunicações, têm taxas de erro mínimas, de modo que erros de transmissão são uma ocorrência rara. Mas outros canais, especialmente links sem fio e loops locais antigos, têm taxas de erro que são comuns e de magnitude maiores. Para esses links, erros de transmissão são a norma. Eles não pode ser evitado a uma despesa ou custo razoável em termos de desempenho. o A conclusão é que os erros de transmissão vieram para ficar. Temos que aprender como lidar com eles.

Os designers de rede desenvolveram duas estratégias básicas para lidar com rors. Ambos adicionam informações redundantes aos dados enviados. Uma estratégia é incluir informações redundantes suficientes para permitir ao receptor deduzir o que os dados transmitidos devem ter sido. A outra é incluir redundância suficiente para permitir que o receptor deduza que ocorreu um erro (mas não qual erro)

e solicite uma retransmissão. A estratégia anterior usa **correção de erros códigos** e o último usa **códigos de detecção de erro**. O uso de correção de erros os códigos costumam ser chamados de **FEC (Correção de erro de encaminhamento)**. Cada uma dessas técnicas ocupa um nicho ecológico diferente. Em canais que são altamente confiáveis, como fibra, é mais barato usar um código de detecção de erros e apenas retransmitir o bloco ocasional considerado defeituoso. No entanto, em canais como links sem fio que cometem muitos erros, é melhor adicionar redundância para cada bloco para que o receptor seja capaz de descobrir o que o originalmente transmitido bloco era. FEC é usado em canais ruidosos porque as retransmissões são tão provavelmente com erro na primeira transmissão.

Uma consideração importante para esses códigos é o tipo de erros que provavelmente ocorrerão cur. Nem os códigos de correção de erros, nem os códigos de detecção de erros podem lidar com todas as possíveis, uma vez que os bits redundantes que oferecem proteção são tão prováveis de serem recebidos erroneamente como os bits de dados (que podem comprometer sua proteção). Seria bom se o canal tratasse os bits redundantes de maneira diferente dos bits de dados, mas não. Eles são apenas bits para o canal. Isso significa que, para evitar erros não detectados rrors, o código deve ser forte o suficiente para lidar com os erros esperados.

Um modelo é que os erros são causados por valores extremos de ruído térmico que sobrecarregam o sinal brevemente e ocasionalmente, dando origem a erros isolados de um único bit. Outro modelo é que os erros tendem a ocorrer em rajadas, em vez de isoladamente. Isto modelo segue a partir dos processos físicos que os geram, como um profundo desaparecer em um canal sem fio ou interferência elétrica transitória em um canal com fio / Ambos os modelos são importantes na prática e têm diferentes compensações. Tendo o erros vêm em rajadas tem vantagens e desvantagens sobre erros de bit. Do lado da vantagem, os dados do computador são sempre enviados em blocos de bits. Suponha que o tamanho do bloco seja de 1000 bits e a taxa de erro seja de 0,001 por bit. E se os erros eram independentes, a maioria dos blocos conteria um erro. Se os erros vieram em rajadas de 100, no entanto, apenas um bloco em 100 seria afetado, em média.

A desvantagem dos erros de burst é que, quando ocorrem, são muito mais difíceis corrigir do que erros isolados.

Outros tipos de erros também existem. Às vezes, a localização de um erro será conhecido, talvez porque a camada física recebeu um sinal analógico que estava longe do valor esperado para 0 ou 1 e declarou que o bit foi perdido. Essa situação é chamado de **canal de eliminação**. É mais fácil corrigir erros em canais de apagamento do que em canais que invertem bits porque mesmo que o valor do bit tenha sido perdido, pelo menos sabemos qual bit está errado. No entanto, muitas vezes não temos o benefício de apagar ures.

Examinaremos os códigos de correção de erros e os códigos de detecção de erros a seguir. No entanto, tenha dois pontos em mente. Primeiro, abordamos esses códigos no link camada porque este é o primeiro lugar em que nos deparamos com o problema da religião habilmente transmitindo grupos de bits. No entanto, os códigos são amplamente usados porque confiabilidade é uma preocupação geral. Códigos de correção de erros também são vistos no corpo camada cal, particularmente para canais ruidosos, e em camadas superiores, particularmente para

Página 228

204

A CAMADA DE LINK DE DADOS INDIVÍDUO. 3

distribuição de mídia e conteúdo em tempo real. Códigos de detecção de erros são comumente usado em camadas de link, rede e transporte.

O segundo ponto a ter em mente é que os códigos de erro são matemática aplicada. A menos que você seja particularmente adepto dos campos de Galois ou das propriedades de matrizes, você deve obter códigos com boas propriedades de uma fonte confiável, em vez do que fazer o seu próprio. Na verdade, é isso que muitos padrões de protocolo fazem, com os mesmos códigos surgindo repetidamente. No material abaixo, estudaremos um código simples em detalhes e, em seguida, descreva resumidamente os códigos avançados. Desta forma, nós

pode entender as compensações do código simples e falar sobre os códigos que são usados na prática por meio dos códigos avançados.

3.2.1 Códigos de correção de erros

Examinaremos quatro códigos de correção de erros diferentes:

1. Códigos de Hamming.
2. Códigos convolucionais binários.
3. Códigos Reed-Solomon.
4. Códigos de verificação de paridade de baixa densidade.

Todos esses códigos adicionam redundância às informações enviadas. Uma moldura de m dados (isto é, mensagem) bits r redundantes (isto é, verificação) bits. Em um **bloco código**, os r bits de verificação são calculados exclusivamente como uma função dos m bits de dados com que estão associados, como se os bits m fossem consultados em uma grande mesa para encontrar seus bits de verificação r correspondentes. Em um **código sistemático**, os m bits de dados são enviados diretamente, junto com os bits de verificação, em vez de serem codificados eles próprios antes que eles sejam enviados. Em um **código linear**, os r bits de verificação são calculados como uma função dos m bits de dados. A adição exclusiva de OR (XOR) ou módulo 2 é um popular. Isso significa que a codificação pode ser feita com operações como matriz multiplicações ou circuitos lógicos simples. Os códigos que veremos nesta seção são códigos de bloco lineares e sistemáticos, a menos que indicado de outra forma.

Seja o comprimento total de um bloco n (isto é, $n = m + r$). Vamos descrever isso como um código (n, m). Uma unidade de n bits contendo dados e bits de verificação é chamada de n -palavra-código de bits. A **taxa de código**, ou simplesmente taxa, é a fração da palavra-código que carrega informações que não são redundantes ou m/n . As taxas usadas na prática variam amplamente. Eles podem ser $1/2$ para um canal barulhento, caso em que metade do recebido as informações são redundantes ou próximas de 1 para um canal de alta qualidade, com apenas um pequeno número de bits de verificação adicionados a uma mensagem grande.

Para entender como os erros podem ser tratados, é necessário primeiro olhar com atenção no que realmente é um erro. Dados quaisquer duas palavras-código que podem ser transmitidas ou recebido - digamos, 10001001 e 10110001 - é possível determinar quantos

Página 229

SEC. 3,2
DETECÇÃO E CORREÇÃO DE ERROS

205

bits correspondentes diferem. Neste caso, 3 bits diferem. Para determinar quantos bits diferirem, apenas XOR as duas palavras-código e contar o número de 1 bits no resultado.

Por exemplo:

10001001
10110001
00111000

O número de posições de bits em que duas palavras-código diferem é chamado de **Hamming distância** (Hamming, 1950). Seu significado é que se duas palavras-código são uma distância de Hamming d separadas, serão necessários d erros de bit único para converter um em outro.

Dado o algoritmo para calcular os bits de verificação, é possível construir uma lista completa das palavras-código legais e, a partir dessa lista, encontrar as duas palavras-código com a menor distância de Hamming. Esta distância é a distância de Hamming de um código completo.

Na maioria das aplicações de transmissão de dados, todas as mensagens de dados possíveis de 2^m são

legal, mas devido à forma como os bits de verificação são calculados, nem todos os 2^n possíveis palavras-código são usadas. Na verdade, quando há r bits de verificação, apenas a pequena fração de $2^m / 2^n$ ou $1 / 2^r$ das mensagens de possíveis palavras de código serão legais. É o dispersão com a qual a mensagem é inserida no espaço de palavras-código que permite que o receptor detecte e corrija os erros.

As propriedades de detecção e correção de erros de um código de bloco dependem de sua distância de Hamming. Para detectar erros d de forma confiável, você precisa de um código de distância $d+1$.

porque com um tal código não há nenhuma maneira que d erros de bit único pode mudar uma palavra-código válida em outra palavra-código válida. Quando o receptor vê um ilegal palavra-código, ele pode dizer que ocorreu um erro de transmissão. Da mesma forma, para corrigir d

erros, você precisa de um código de distância $2d + 1$ porque dessa forma as palavras-código legais são tão distantes que, mesmo com d mudanças, a palavra-código original está ainda mais próxima do que qualquer outra palavra-código. Isso significa que a palavra-código original pode ser determinada de maneira única

com base na suposição de que um número maior de erros é menos provável.

Como um exemplo simples de código de correção de erros, considere um código com apenas quatro palavras-código válidas:

0000000000, 0000011111, 1111100000 e 1111111111

Este código tem uma distância de 5, o que significa que pode corrigir erros duplos ou detectar erros quádruplos. Se a palavra-código 0000000111 chegar e esperarmos apenas erros de bit único ou duplo, o receptor saberá que o original deve ter

sido 0000011111. Se, no entanto, um erro triplo alterar 0000000000 para

0000000111, o erro não será corrigido corretamente. Alternativamente, se esperarmos todos esses erros, podemos detectá-los. Nenhuma das palavras-código recebidas é legal

palavras-código, então um erro deve ter ocorrido. Deve ser evidente que neste ex-

emplo, não podemos corrigir erros duplos e detectar erros quádruplos porque

isso exigiria que interpretássemos uma palavra-código recebida de duas maneiras diferentes.

Página 230

206

A CAMADA DE LINK DE DADOS

INDIVÍDUO. 3

Em nosso exemplo, a tarefa de decodificar encontrando a palavra-código legal que é mais próximo da palavra-código recebida pode ser feito por inspeção. Infelizmente, no caso mais geral em que todas as palavras-código precisam ser avaliadas como candidatas, este tarefa pode ser uma pesquisa demorada. Em vez disso, os códigos práticos são projetados para que eles admitem atalhos para encontrar o que provavelmente era a palavra-código original.

Imagine que queremos projetar um código com m bits de mensagem e r bits de verificação isso permitirá que todos os erros individuais sejam corrigidos. Cada uma das 2^{m+r} mensagens legais tem

n palavras-código ilegais a uma distância de 1 dele. Estes são formados por sistematicamente inverter cada um dos n bits na palavra-código de n bits formada a partir dele. Assim, cada um dos as mensagens legais de 2^m requerem padrões de $n+1$ bits dedicados a ela. Desde o total o número de padrões de bits é 2^n , devemos ter $(n+1)2^m \leq 2^n$. Usando $n = m+r$, este exigência torna-se

$$(m+r+1) \leq 2^r$$

(3-1)

Dado m , isso coloca um limite inferior no número de bits de verificação necessários para corrigir sen-
erros gle.

Este limite inferior teórico pode, de fato, ser alcançado usando um método devido a Hamming (1950). Nos **códigos de Hamming**, os bits da palavra-código são numerados consecutivamente, começando com o bit 1 na extremidade esquerda, o bit 2 na sua direita imediata e assim

em. Os bits com potências de 2 (1, 2, 4, 8, 16, etc.) são bits de verificação. O resto (3, 5, 6, 7, 9, etc.) são preenchidos com os bits de dados m . Este padrão é mostrado para um (11,7) Código de Hamming com 7 bits de dados e 4 bits de verificação na Fig. 3-6. Cada bit de verificação

força a soma do módulo 2, ou paridade, de alguma coleção de bits, incluindo ela mesma, para ser par (ou ímpar). Um bit pode ser incluído em vários cálculos de bit de verificação. Para veja para quais bits de verificação o bit de dados na posição k contribui, reescreva k como uma soma de

potências de 2. Por exemplo, $11 = 1 + 2 + 8$ e $29 = 1 + 4 + 8 + 16$. Um bit é verificado apenas por aqueles bits de verificação que ocorrem em sua expansão (por exemplo, o bit 11 é verificado

pelos bits 1, 2 e 8). No exemplo, os bits de verificação são calculados para paridade par somas para uma mensagem que é a letra ASCII " A. "

```
Enviei
palavra-código
Recebido
palavra-código
0 0 1 0 0 0 0 1 0 0 1
p1 p2 m3 p4 m5 m6 m7 p8 m9 m10 m11
Verifica
bits
Canal
0 0 1 0 1 0 0 1 0 0 1
1 bit
erro
Síndrome
0101
Verifica
resultados
UMA
1000001
Giro
bit 5
UMA
1000001
mensagem
mensagem
```

Figura 3-6. Exemplo de um código de Hamming (11, 7) corrigindo um erro de bit único.

Esta construção fornece um código com uma distância de Hamming de 3, o que significa que pode corrigir erros únicos (ou detectar erros duplos). O motivo do próprio a numeração cuidadosa da mensagem e os bits de verificação tornam-se aparentes na decodificação

Página 231

SEC. 3,2

DETECÇÃO E CORREÇÃO DE ERROS

207

processo. Quando uma palavra-código chega, o receptor refaz o cálculo do bit de verificação incluindo os valores dos bits de verificação recebidos. Chamamos isso de cheque resultados. Se os bits de verificação estiverem corretos, então, para somas de paridade pares, cada resultado de verificação

deve ser zero. Nesse caso, a palavra-código é aceita como válida.

Se os resultados da verificação não forem todos zero, entretanto, um erro foi detectado. o conjunto de resultados de verificação forma a **síndrome do erro** que é usada para localizar e corrigir o erro. Na Fig. 3-6, ocorreu um erro de bit único no canal, então a verificação os resultados são 0, 1, 0 e 1 para $k = 8, 4, 2$ e 1, respectivamente. Isso dá uma síndrome de 0101 ou $4 + 1 = 5$. Pelo projeto do esquema, isso significa que o quinto bit está em erro. Inverter o bit incorreto (que pode ser um bit de verificação ou um bit de dados) e discardar os bits de verificação dá a mensagem correta de um ASCII " A. "

As distâncias de Hamming são valiosas para a compreensão dos códigos de bloco, e Hamming os códigos de bloco são usados na memória de correção de erros. No entanto, a maioria das redes usa códigos mais fortes. O segundo código que veremos é um **código convolucional**. Isto código é o único que abordaremos que não é um código de bloco. Em uma convolucional código, um codificador processa uma sequência de bits de entrada e gera uma sequência de bits de saída. Não há tamanho de mensagem natural ou limite de codificação como em um bloco código. A saída depende dos bits de entrada atuais e anteriores. Ou seja, o codificador tem memória. O número de bits anteriores dos quais a saída depende é chamado de **comprimento de restrição** do código. Os códigos convolucionais são especificados em termos de sua taxa e comprimento de restrição.

Os códigos convolucionais são amplamente utilizados em redes implantadas, por exemplo, como parte do sistema de telefonia móvel GSM, em comunicações por satélite e em 802.11.

Como exemplo, um código convolucional popular é mostrado na Figura 3-7. Este código é conhecido como o código convolucional NASA de $R = 1/2$ e $k = 7$, desde que foi primeiro usado para as missões espaciais Voyager começando em 1977. Desde então, tem sido generosamente reutilizado, por exemplo, como parte do 802.11.

```
Entrada
mordeu
Resultado
bit 1
```

S 1
S 2
S 3
S 4
S 5
S 6
Resultado
bit 2

Figura 3-7. O código convolucional binário da NASA usado em 802.11.

Na Fig. 3-7, cada bit de entrada no lado esquerdo produz dois bits de saída no lado direito que são somas XOR do estado de entrada e interno. Uma vez que trata com bits e executa operações lineares, este é um binário, convolucional linear código. Uma vez que 1 bit de entrada produz 2 bits de saída, a taxa de código é 1 / 2. Não é sistema temático, pois nenhum dos bits de saída é simplesmente o bit de entrada.

Página 232

208

A CAMADA DE LINK DE DADOS INDIVÍDUO. 3

O estado interno é mantido em seis registros de memória. Cada vez que outro bit está dentro colocar os valores nos registros são deslocados para a direita. Por exemplo, se 111 é uma entrada e o estado inicial é todo zeros, o estado interno, escrito da esquerda para a direita, se tornará 100000, 110000 e 111000 após o primeiro, segundo e terceiro bits terem sido inseridos.

Os bits de saída serão 11, seguidos por 10 e, em seguida, 01. São necessários sete turnos para esvaziar uma entrada completamente para que não afete a saída. A restrição o comprimento desse código é, portanto, $k = 7$.

Um código convolucional é decodificado encontrando a sequência de bits de entrada que é mais provável de ter produzido a sequência observada de bits de saída (que inclui quaisquer erros). Para pequenos valores de k , isso é feito com um algoritmo amplamente utilizado de-

velado por Viterbi (Forney, 1973). O algoritmo percorre a sequência observada, mantendo para cada etapa e para cada estado interno possível a sequência de entrada que teria produzido a sequência observada com o menor número de erros. A entrada se-

A sequência que requer menos erros no final é a mensagem mais provável.

Os códigos convolucionais têm sido populares na prática porque é fácil de fatorar a incerteza de um bit ser 0 ou 1 na decodificação. Por exemplo, suponha -1V é o nível lógico 0 e +1V é o nível lógico 1, podemos receber 0,9V e -0,1V para 2 bits. Em vez de mapear esses sinais para 1 e 0 imediatamente, nós gostaria de tratar 0,9V como "muito provavelmente 1" e -0,1V como "talvez 0" e corretifique a sequência como um todo. Extensões do algoritmo de Viterbi podem funcionar com essas incertezas para fornecer uma correção de erro mais forte. Esta abordagem de trabalho com a incerteza de um bit é chamada de **decodificação de decisão suave**. Por outro lado, decidir ver se cada bit é 0 ou 1 antes que a correção de erro subsequente seja chamada **decodificação de difícil decisão**.

O terceiro tipo de código de correção de erros que descreveremos é o **Reed-Solomon código**. Como os códigos de Hamming, os códigos de Reed-Solomon são códigos de blocos lineares e

muitas vezes também são sistemáticos. Ao contrário dos códigos de Hamming, que operam em indivíduos

bits ual, códigos Reed-Solomon operam em símbolos de bits m . Naturalmente, o matemati-

Os ics estão mais envolvidos, portanto, descreveremos sua operação por analogia.

Os códigos Reed-Solomon são baseados no fato de que todo polinômio de n graus é determinado exclusivamente por $n + 1$ pontos. Por exemplo, uma linha com a forma $ax + b$ é determinado por dois pontos. Os pontos extras na mesma linha são redundantes, o que é útil para correção de erros. Imagine que temos dois pontos de dados que representam um linha e enviamos esses dois pontos de dados mais dois pontos de verificação escolhidos para estar na mesma linha. Se um dos pontos for recebido com erro, ainda podemos recuperar os dados pontos ajustando uma linha aos pontos recebidos. Três dos pontos estarão no linha, e um ponto, o que está com erro, não. Ao encontrar a linha, temos corrigiu o erro.

Os códigos Reed-Solomon são na verdade definidos como polinômios que operam sobre campos finitos, mas funcionam de maneira semelhante. Para símbolos de bits m , as palavras-código têm $2^m - 1$ símbolos de comprimento. Uma escolha popular é fazer $m = 8$ para que os símbolos sejam bytes. Uma palavra-código tem então 255 bytes. O código (255, 233) é amplamente utilizado; isto adiciona 32 símbolos redundantes a 233 símbolos de dados. Decodificação com correção de erros

Página 233

SEC. 3,2
DETECÇÃO E CORREÇÃO DE ERROS

209

é feito com um algoritmo desenvolvido por Berlekamp e Massey que pode executar adequadamente a tarefa de ajuste para códigos de comprimento moderado (Massey, 1969). Os códigos Reed-Solomon são amplamente usados na prática por causa de sua forte propriedades de correção de erros, especialmente para erros de explosão. Eles são usados para DSL, dados por cabo, comunicações por satélite e talvez de forma mais onipresente em CDs, DVDs e discos Blu-ray. Porque eles são baseados em símbolos de m bits, um único bit erro e um erro de burst de m bits são tratados simplesmente como um erro de símbolo. Quando $2t$ símbolos redundantes são adicionados, um código Reed-Solomon é capaz de corrigir até t erros em qualquer um dos símbolos transmitidos. Isso significa, por exemplo, que o (255, 233) código, que tem 32 símbolos redundantes, pode corrigir até 16 erros de símbolo. Uma vez que os símbolos podem ser consecutivos e têm 8 bits a cada um, uma rajada de erro de até 128 bits podem ser corrigidos. A situação é ainda melhor se o modelo de erro for uma das rasuras (por exemplo, um arranhão em um CD que oblitera alguns símbolos). Nisso caso, até $2t$ erros podem ser corrigidos.

Os códigos Reed-Solomon são frequentemente usados em combinação com outros códigos, como um código convolucional. O pensamento é o seguinte. Códigos convolucionais são eficazes em lidar com erros de bits isolados, mas eles irão falhar, provavelmente com uma explosão de erros, se há muitos erros no fluxo de bits recebido. Adicionando um Reed-Solomon código dentro do código convolucional, a decodificação Reed-Solomon pode limpar o rascunho de erro, uma tarefa em que é muito bom. O código geral fornece boas proteção contra erros simples e burst.

O código final de correção de erros que abordaremos é o **LDPC** (Low-Density Código de verificação de paridade). Os códigos LDPC são códigos de blocos lineares que foram inventados por Robert Gallagher em sua tese de doutorado (Gallagher, 1962). Como a maioria das teses, eles foram prontamente esquecidos, apenas para serem reinventados em 1995, quando os avanços na computação poder ingente os tornou práticos.

Em um código LDPC, cada bit de saída é formado a partir de apenas uma fração da entrada bits. Isso leva a uma representação de matriz do código que tem uma densidade baixa de 1s, daí o nome do código. As palavras-código recebidas são decodificadas com um algoritmo de aproximação que melhora iterativamente em um melhor ajuste do recebido dados a uma palavra-código legal. Isso corrige os erros.

Os códigos LDPC são práticos para grandes tamanhos de bloco e têm excelente correção de erro habilidades de correção que superam muitos outros códigos (incluindo os que temos olhado) na prática. Por esta razão, eles estão sendo rapidamente incluídos em novos protocolos. Eles fazem parte do padrão para transmissão de vídeo digital, 10 Gbps Ethernet, redes de linha de energia e a versão mais recente de 802.11. Espere ver mais deles em redes futuras.

3.2.2 Códigos de detecção de erros

Códigos de correção de erros são amplamente usados em links sem fio, que são notórios muito barulhento e sujeito a erros quando comparado às fibras ópticas. Sem erro-corrigindo códigos, seria difícil fazer qualquer coisa. No entanto, sobre fibra ou

210**A CAMADA DE LINK DE DADOS
INDIVÍDUO. 3**

cobre de alta qualidade, a taxa de erro é muito menor, portanto, detecção e retransmissão de erros missão é geralmente mais eficiente lá para lidar com o erro ocasional.

Examinaremos três códigos diferentes de detecção de erros. Eles são todos lineares, códigos de bloqueio sistemáticos:

1. Paridade.

2. Soma de verificação.

3. Verificações de redundância cíclica (CRCs).

Para ver como eles podem ser mais eficientes do que códigos de correção de erros, considere o primeiro código de detecção de erro, no qual um único **bit de paridade** é anexado aos dados. O bit de paridade é escolhido de forma que o número de 1 bits na palavra-código seja par (ou ímpar). Fazer isso é equivalente a calcular o bit de paridade (par) como o módulo 2 soma ou XOR dos bits de dados. Por exemplo, quando 1011010 é enviado em paridade par, um bit é adicionado ao final para torná-lo 10110100. Com paridade ímpar 1011010 torna-se 10110101. Um código com um único bit de paridade tem uma distância de 2, uma vez que qualquer bit único

erro produz uma palavra-código com a paridade errada. Isso significa que ele pode detectar erros de bit único.

Considere um canal no qual os erros são isolados e a taxa de erro é 10⁻⁶

por

mordeu. Isso pode parecer uma pequena taxa de erro, mas é, na melhor das hipóteses, uma taxa justa para uma longa

cabo que é um desafio para a detecção de erros. Links de LAN típicos fornecem erro de bit taxas de 10⁻¹⁰.

Deixe o tamanho do bloco ser 1000 bits. Para fornecer correção de erros para Blocos de 1000 bits, sabemos da Eq. (3-1) que 10 bits de verificação são necessários. Assim, um megabit de dados exigiria 10.000 bits de verificação. Para simplesmente detectar um bloco com um erro único de 1 bit, um bit de paridade por bloco será suficiente. Uma vez a cada 1000 blocos, um bloco será encontrado com erro e um bloco extra (1001 bits) terá que ser transmitido para reparar o erro. A sobrecarga total para a detecção de erros e re-método de transmissão é de apenas 2001 bits por megabit de dados, contra 10.000 bits para um Código de Hamming.

Uma dificuldade com este esquema é que um único bit de paridade só pode ser confiável detectar um erro de bit único no bloco. Se o bloco estiver muito distorcido por uma longa explosão erro, a probabilidade de que o erro seja detectado é de apenas 0,5, o que dificilmente é aceitável. As chances podem ser melhoradas consideravelmente se cada bloco a ser enviado for considerada como uma matriz retangular com n bits de largura ek bits de altura. Agora, se calcularmos

e enviar um bit de paridade para cada linha, até k erros de bits serão detectados de forma confiável como

desde que haja no máximo um erro por linha.

No entanto, há outra coisa que podemos fazer que oferece melhor proteção contra erros de explosão: podemos calcular os bits de paridade sobre os dados em um ordem diferente da ordem em que os bits de dados são transmitidos. Fazer isso é chamado **intercalação**. Neste caso, vamos calcular um bit de paridade para cada uma das n colunas e enviar todos os bits de dados como k linhas, enviando as linhas de cima para baixo e os bits em cada linha da esquerda para a direita da maneira usual. Na última linha, enviamos os n bits de paridade. Essa ordem de transmissão é mostrada na Fig. 3-8 para $n = 7$ $ek = 7$.

211

```
Estouro  
erro  
Canal  
Transmite  
ordem  
Bits de paridade  
1011110  
N  
c  
eu  
W  
o  
r  
k  
Erros de paridade  
1011110  
N  
e  
t  
W  
o  
r  
k  
1001110  
1100101  
1110100  
1110111  
1101111  
1110010  
1101011  
1001110  
1100011  
1101100  
1110111  
1101111  
1110010  
1101011  
1101011
```

Figura 3-8. Intercalação de bits de paridade para detectar um erro de burst.

A intercalação é uma técnica geral para converter um código que detecta (ou corrige) erros isolados em um código que detecta (ou corrige) erros de burst. Na Fig. 3-8, quando ocorre um erro de rajada de comprimento $n = 7$, os bits que estão em erro são espalhados por diferentes colunas. (Um erro de explosão não significa que todos os bits estão errados; apenas implica que pelo menos o primeiro e o último estão errados. Na Fig. 3-8, 4 bits foram invertidos em um intervalo de 7 bits.) No máximo 1 bit em cada uma das n colunas será afetado, portanto os bits de paridade nessas colunas detectarão o erro. Este método usa n paridade bits em blocos de bits de dados kn para detectar um único erro de rajada de comprimento n ou menos.

Uma rajada de comprimento $n + 1$ passará sem ser detectada, no entanto, se o primeiro bit for invertido, o último bit é invertido e todos os outros bits estão corretos. Se o bloco for muito distorcida por uma longa explosão ou por várias explosões mais curtas, a probabilidade de que qualquer das n colunas terão a paridade correta por acidente é 0,5, então a probabilidade de um bloco inválido sendo aceito quando não deveria ser 2⁻ⁿ.

O segundo tipo de código de detecção de erros, a **soma de verificação**, está intimamente relacionado com grupos de bits de paridade. A palavra "checksum" é freqüentemente usada para significar um grupo de

verifique os bits associados a uma mensagem, independentemente de como são calculados. Um grupo

de bits de paridade é um exemplo de soma de verificação. No entanto, existem outros, mais fortes checksums com base em uma soma contínua dos bits de dados da mensagem. A soma de verificação é geralmente colocado no final da mensagem, como o complemento da função de somação. Dessa forma, os erros podem ser detectados pela soma de toda a palavra-código recebida, bits de dados e soma de verificação. Se o resultado for zero, nenhum erro foi detectado.

Um exemplo de checksum é o checksum da Internet de 16 bits usado em todos os pacotes de rede como parte do protocolo IP (Braden et al., 1988). Este checksum é um soma dos bits da mensagem dividida em palavras de 16 bits. Porque este método opera

em palavras em vez de em bits, como na paridade, erros que deixam a paridade inalterada ainda pode alterar a soma e ser detectado. Por exemplo, se o bit de ordem mais baixa em dois palavras diferentes são trocadas de 0 a 1, uma verificação de paridade entre esses bits falha em detectar um erro. No entanto, dois 1s serão adicionados à soma de verificação de 16 bits para produzir um resultado diferente. O erro pode então ser detectado.

Página 236

212

A CAMADA DE LINK DE DADOS

INDIVÍDUO. 3

A soma de verificação da Internet é calculada na aritmética do complemento de algum, em vez de como a soma do módulo 2¹⁶. Na aritmética do complemento de algum, um número negativo é o complemento bit a bit de sua contraparte positiva. Os computadores modernos funcionam com dois complemento aritmética, em que um número negativo é o complemento de um mais

1. Em um computador complemento de dois, a soma do complemento de um é equivalente para pegar o módulo de soma 2¹⁶ e adicionar qualquer estouro dos bits de ordem superior de volta nos bits de ordem inferior. Este algoritmo oferece uma cobertura mais uniforme dos dados pelos bits de soma de verificação. Caso contrário, dois bits de ordem superior podem ser adicionados, estouro e

ser perdido sem alterar a soma. Há outro benefício também. Completou-se
mento tem duas representações de zero, todos 0s e todos 1s. Isso permite um valor (por exemplo,
todos os 0s) para indicar que não há soma de verificação, sem a necessidade de outro campo.

Por décadas, sempre foi assumido que os quadros devem ter a soma de verificação
conter bits aleatórios. Todas as análises de algoritmos de soma de verificação foram feitas sob este
suposição. Inspeção de dados reais por Partridge et al. (1995) mostrou isso como-
suposição estar completamente errada. Como consequência, erros não detectados são em alguns
casos muito mais comuns do que se pensava anteriormente.

A soma de verificação da Internet em particular é eficiente e simples, mas fornece
proteção em alguns casos precisamente porque é uma soma simples. Não detecta
a exclusão ou adição de zero dados, nem troca de partes da mensagem, e
fornecer proteção fraca contra emendas de mensagens em que partes de dois pacotes
são colocados juntos. Esses erros podem parecer muito improváveis de ocorrer por proc-
esses, mas eles são apenas o tipo de erro que pode ocorrer com hardware com bugs.

Uma escolha melhor é a **soma de verificação** de Fletcher (Fletcher, 1982). Inclui uma posição
componente tradicional, adicionando o produto dos dados e sua posição ao
soma. Isso fornece uma detecção mais forte de alterações na posição dos dados.

Embora os dois esquemas anteriores possam às vezes ser adequados em
camadas, na prática, um terceiro e mais forte tipo de código de detecção de erros está
difundir o uso na camada de enlace: o **CRC (Cyclic Redundancy Check)**, também conhecido
como um **código polinomial**. Os códigos polinomiais são baseados no tratamento de cadeias de bits
como

representações de polinômios com coeficientes de 0 e 1 apenas. Um quadro k -bit é
considerada como a lista de coeficientes para um polinômio com k termos, variando de x^{k-1}
a x^0 . Esse polinômio é considerado de grau $k-1$. A ordem superior (extrema esquerda)
bit é o coeficiente de x^{k-1}
, o próximo bit é o coeficiente de x^{k-2}
, e assim por diante.

Por exemplo, 110001 tem 6 bits e, portanto, representa um polinômio de seis termos com
coeficientes 1, 1, 0, 0, 0 e 1: $1x^5 + 1x^4 + 0x^3 + 0x^2 + 0x^1 + 1x^0$.

A aritmética polinomial é feita módulo 2, de acordo com as regras da álgebra
teoria de campo. Não possui carrega para adição ou empresta para subtração. Ambos
adição e subtração são idênticas ao OU exclusivo. Por exemplo:

10011011

00110011

11110000

01010101

```

+ 11001010
+ 11001101
- 10100110
- 10101111
01010001
11111110
01010110
11111010

```

A divisão longa é realizada exatamente da mesma maneira que em binário, exceto que

SEC. 3,2

DETECÇÃO E CORREÇÃO DE ERROS

213

a subtração é feita novamente módulo 2. Diz-se que um divisor "entra em" um dividendo se o dividendo tiver tantos bits quanto o divisor.

Quando o método do código polinomial é empregado, o remetente e o receptor devem concordar sobre um **polinômio gerador**, $G(x)$, com antecedência. Tanto o alto quanto o baixo bits de pedido do gerador devem ser 1. Para calcular o CRC para algum quadro com bits m correspondentes ao polinômio $M(x)$, o quadro deve ser maior que o polinômio gerador. A ideia é anexar um CRC ao final do quadro em forma que o polinômio representado pelo quadro de soma de verificação é dividido por $G(x)$. Quando o receptor obtém o quadro de soma de verificação, ele tenta dividir-lo por $G(x)$. Se houver um resto, houve um erro de transmissão.

O algoritmo para calcular o CRC é o seguinte:

1. Seja r o grau de $G(x)$. Anexar r bits de zero ao final de ordem inferior do quadro, portanto, agora contém $m + r$ bits e corresponde ao polinômio $x^r M(x)$.
2. Divida a sequência de bits correspondente a $G(x)$ na sequência de bits correspondente ponderando a $x^r M(x)$, usando a divisão do módulo 2.
3. Subtraia o restante (que é sempre r ou menos bits) do bit string correspondente $x^r M(x)$ usando a subtração do módulo 2. Lá-sult é o quadro de soma de verificação a ser transmitido. Chame seu polinômio $T(x)$.

A Figura 3-9 ilustra o cálculo para um quadro 1101011111 usando o gerador $G(x) = x^4 + x + 1$.

Deve ficar claro que $T(x)$ é divisível (módulo 2) por $G(x)$. Em qualquer divisão problema, se você diminuir o dividendo pelo restante, o que sobra é a divisão ble pelo divisor. Por exemplo, na base 10, se você dividir 210.278 por 10.941, o resto é 2399. Se você subtrair 2399 de 210.278, o que sobra (207.879) é divisível por 10.941.

Agora, vamos analisar o poder desse método. Que tipos de erros serão de- protegido? Imagine que ocorre um erro de transmissão, de modo que, em vez da string de bits para $T(x)$ chegando, $T(x) + E(x)$ chega. Cada 1 bit em $E(x)$ corresponde a um bit que foi invertido. Se houver k 1 bits em $E(x)$, k erros de bit único ocorreram.

Um único erro de burst é caracterizado por um 1 inicial, uma mistura de 0s e 1s, e um final 1, com todos os outros bits sendo 0.

Ao receber o quadro de soma de verificação, o receptor o divide por $G(x)$; aquele ou seja, ele calcula $[T(x) + E(x)] / G(x)$. $T(x) / G(x)$ é 0, então o resultado do cálculo ção é simplesmente $E(x) / G(x)$. Aqueles erros que correspondem a polinômios contendo $G(x)$ como um fator passará despercebido; todos os outros erros serão detectados. Se houve um erro de bit único, $E(x) = x^i$, onde i determina qual bit é em erro. Se $G(x)$ contém dois ou mais termos, ele nunca vai se dividir em $E(x)$, então todos erros de bit único serão detectados.

```

00011
01001
11001
1
01011
11001
1
01111
11001
1
00000
11110
1
11
0
11100
00000
110
0 0
000
0 0
0
10
0 0
0
00
0 0
0
1
0
1
1
0
1
0
1
1
0
01 0
0
0000 0
1000 0
1 0
1 1
1
1 0 0
1
Restante
Quociente (jogado fora)
Quadro com quatro zeros anexados

```

```

0
00
0
0
1
11111
1
11

```

Quadro com quatro zeros anexados
menos resto

Quadro transmitido:
 1
 1 1 0
 1 0 0
 0 1 1 1 1 1

Quadro, Armação:
 1 1
 1 1 0 0 0 0 1 1 1 0

Gerador:

Figura 3-9. Exemplo de cálculo do CRC.

Se houver dois erros isolados de bit único, $E(x) = x_i + x_j$, onde $i > j$.

Alternativamente, isso pode ser escrito como $E(x) = x_j(x_{i-j} + 1)$. Se assumirmos que $G(x)$ não é divisível por x , uma condição suficiente para que todos os erros duplos sejam detectados é que $G(x)$ não divide $x_k + 1$ por qualquer k até o valor máximo de $i - j$ (ou seja, até o comprimento máximo do quadro). Polinômios simples de baixo grau que fornecem proteção para quadros longos são conhecidos. Por exemplo, $x^{15} + x^{14} + 1$ não vai dividir $x_k + 1$ para qualquer valor de k abaixo de 32.768.

Se houver um número ímpar de bits com erro, $E(X)$ contém um número ímpar de

termos (por exemplo, $x^5 + x^2 + 1$, mas não $x^2 + 1$). Curiosamente, nenhum polinômio com um estranho

número de termos tem $x + 1$ como um fator no sistema módulo 2. Fazendo $x + 1$ a fator de $G(x)$, podemos capturar todos os erros com um número ímpar de bits invertidos. Finalmente, e mais importante, um código polinomial com r bits de verificação detectará todos erros de explosão de comprimento $\leq r$. Um erro de ruptura de comprimento k pode ser representado por

$x^i (x^{k-1} + \dots + 1)$, onde i determina a que distância da extremidade direita do quadro ceived onde o burst está localizado. Se $G(x)$ contém um termo x^0 , ele não terá x^i como um fator, então, se o grau da expressão entre parênteses for menor que o grau de $G(x)$, o resto nunca pode ser zero.

Página 239

SEC. 3.2

DETECÇÃO E CORREÇÃO DE ERROS

215

Se o comprimento da rajada for $r + 1$, o restante da divisão por $G(x)$ será zero se e somente se o burst for idêntico a $G(x)$. Por definição de explosão, o primeiro e os últimos bits devem ser 1, portanto, sua correspondência depende dos $r - 1$ bits intermediários. Se todas as combinações forem consideradas igualmente prováveis, a probabilidade de tal incorreto frame reto aceito como válido é $\frac{1}{2^{r-1}}$.

Também pode ser mostrado que quando ocorre uma rajada de erro maior que $r + 1$ bits ou quando várias rajadas mais curtas ocorrem, a probabilidade de um quadro ruim passar despercebido é $\frac{1}{2^r}$, assumindo que todos os padrões de bits são igualmente prováveis.

Certos polinômios se tornaram padrões internacionais. O usado em

IEEE 802 seguiu o exemplo de Ethernet e é

$$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x^1 + 1$$

Entre outras propriedades desejáveis, tem a propriedade de detectar todas as explosões de comprimento de 32 ou menos e todos os bursts afetando um número ímpar de bits. Tem sido usado amplamente desde os anos 1980. No entanto, isso não significa que seja a melhor escolha. Usando uma pesquisa computacional exaustiva, Castagnoli et al. (1993) e Koopman (2002) encontraram os melhores CRCs. Esses CRCs têm uma distância de Hamming de 6 para tamanhos de mensagens, enquanto o padrão IEEE CRC-32 tem uma distância de Hamming de apenas

4 -

Embora o cálculo necessário para calcular o CRC possa parecer complicado ed, é fácil calcular e verificar CRCs no hardware com registro de deslocamento simples circuitos (Peterson e Brown, 1961). Na prática, este hardware é quase sempre usava. Dezenas de padrões de rede incluem vários CRCs, incluindo virtualmente todas as LANs (por exemplo, Ethernet, 802.11) e links ponto a ponto (por exemplo, pacotes sobre SONET).

3.3 PROTOCOLOS DE LINK DE DADOS ELEMENTARES

Para introduzir o assunto dos protocolos, vamos começar examinando três protocolos de complexidade crescente. Para leitores interessados, um simulador para estes e os protocolos subsequentes estão disponíveis na Web (consulte o prefácio). Antes de olharmos nos protocolos, é útil tornar explícitas algumas das suposições subjacentes ao modelo de comunicação.

Para começar, assumimos que a camada física, camada de enlace de dados e rede camada são processos independentes que se comunicam passando mensagens de volta e adiante. Uma implementação comum é mostrada na Figura 3-10. A camada física processamental e alguns dos processos da camada de enlace executados em hardware dedicado chamado **NIC**

(**Placa de interface de rede**). O resto do processo da camada de link e a rede processo de camada executado na CPU principal como parte do sistema operacional, com o software para o processo da camada de link geralmente assumindo a forma de um **driver de dispositivo**. Howev-

er, outras implementações também são possíveis (por exemplo, três processos descarregados para dedicado hardware chamado de **acelerador de rede**, ou três processos em execução no

216

A CAMADA DE LINK DE DADOS

INDIVÍDUO. 3

CPU principal em uma proporção definida por software). Na verdade, a implementação preferida muda de década para década com compensações de tecnologia. Em qualquer caso, tratando as três camadas como processos separados tornam a discussão conceitualmente mais limpa e também serve para enfatizar a independência das camadas.

Rede
Cabo (médio)
PHY
Ligação
Ligaçao
Inscrição
Interface de rede
Cartão (NIC)
Motorista
Sistema operacional
Computador

Figura 3-10. Implementação das camadas físicas, de link de dados e de rede.

Outra suposição fundamental é que a máquina *A* deseja enviar um longo fluxo de dados para a máquina *B*, usando um serviço confiável e orientado para a conexão. Mais tarde, vamos considerar

o caso em que *B* também deseja enviar dados para *A* simultaneamente. *A* é assumido como tem um suprimento infinito de dados prontos para enviar e nunca tem que esperar que os dados sejam

produzido. Em vez disso, quando *A* ‘s camada de enlace de dados pede dados, a camada de rede é sempre capazes de cumprir imediatamente. (Essa restrição também será eliminada posteriormente.) Também assumimos que as máquinas não travam. Ou seja, esses protocolos lidam com erros de comunicação, mas não os problemas causados por falhas de computador e reinicializando.

No que diz respeito à camada de enlace de dados, o pacote passou pelo interface a ele da camada de rede são dados puros, cujo cada bit deve ser entregue a camada de rede do destino. O fato de que a camada de rede do destino pode interpretar parte do pacote como um cabeçalho não interessa à camada de enlace de dados.

Quando a camada de enlace de dados aceita um pacote, ela encapsula o pacote em um quadro adicionando um cabeçalho de link de dados e um trailer a ele (consulte a Fig. 3-1). Assim, um quadro

consiste em um pacote incorporado, algumas informações de controle (no cabeçalho) e um checksum (no trailer). O quadro é então transmitido para a camada de enlace em outra máquina. Assumiremos que existem procedimentos de biblioteca adequados

para a camada física para enviar um quadro e da camada física para receber um quadro. Esses procedimentos calculam e acrescentam ou verificam a soma de verificação (que geralmente é feito em hardware) para que não precisemos nos preocupar com isso como parte dos protocolos desenvolvemos nesta seção. Eles podem usar o algoritmo CRC discutido na seção anterior, por exemplo.

Inicialmente, o receptor não tem nada a fazer. Ele apenas fica esperando por alguma coisa a acontecer. Nos protocolos de exemplo ao longo deste capítulo, iremos indicar que a camada de enlace de dados está esperando que algo aconteça pela chamada de procedimento

SEC. 3,3

PROTÓCOLOS DE LINK DE DADOS ELEMENTARES

217

```
# define MAX PKT 1024
```

```
/* determina o tamanho do pacote em bytes */
```

```

typedef enum {false, true} booleano;
/* tipo booleano */
typedef unsigned int seq_nr;
/* sequência ou números de confirmação */

typedef struct {dados de char não assinados [MAX_PKT];} pacote; /* definição de pacote */
typedef enum {data, ack, nak} tipo_de_quadro;
/* definição de tipo de quadro */

typedef struct {
    /* OS quadros são transportados nesta camada */
    tipo_de_moldura;
    /* que tipo de moldura é? */
    seq_nr seq;
    /* número de sequência */
    seq_nr ack;
    /* número de confirmação */
    informações do pacote;
    /* o pacote da camada de rede */
    } quadro, Armação;

/* Esperar que um evento aconteça; retorna seu tipo no evento. */
nula espera por evento (tipo de evento * evento);

/* Busca um pacote da camada de rede para transmissão no canal. */
vazio da camada de rede (pacote * p);

/* Entrega informações de um quadro de entrada para a camada de rede. */
vazio para a camada de rede (pacote * p);

/* Vá obter um quadro de entrada da camada física e copie-o para r. */
vazio da camada física (frame * r);

/* Passe o quadro para a camada física para transmissão. */
vazio para a camada física (frame * s);

/* Inicia o relógio e ativa o evento de tempo limite. */
void start_timer (seq_nr k);

/* Pare o relógio e desabilite o evento de tempo limite. */
void stop_timer (seq_nr k);

/* Inicia um temporizador auxiliar e habilita o evento de tempo limite de reconhecimento. */
void start_ack_timer (void);

/* Pare o cronômetro auxiliar e desabilite o evento de tempo limite de reconhecimento. */
void stop_ack_timer (void);

/* Permite que a camada de rede cause um evento pronto para a camada de rede. */
void permite_a_camada_de_rede (void);

/* Proíbe a camada de rede de causar um evento pronto para a camada de rede. */
void desativar_camada_de_rede (void);

/* Macro inc é expandido em linha: incrementa k circularmente. */
#define inc (k) se (k < MAX_SEQ) k = k + 1; senão k = 0

```

Figura 3-11. Algumas definições necessárias nos protocolos a seguir. Estes definições estão localizadas no arquivo *protocol.h*.

aguarde o evento (*e evento*). Este procedimento só retorna quando algo aconteceu escrito (por exemplo, chegou um quadro). Ao retornar, o *evento* variável diz o que aconteceu escrito. O conjunto de eventos possíveis difere para os vários protocolos a serem descritos e será definido separadamente para cada protocolo. Observe que de uma forma mais realista situação, a camada de enlace de dados não ficará em um loop apertado esperando por um evento, como nós sugeriram, mas receberá uma interrupção, o que fará com que pare qualquer coisa estava fazendo e vá lidar com o quadro de entrada. No entanto, para simplificar, nós irá ignorar todos os detalhes da atividade paralela dentro da camada de enlace e assumir

que é dedicado em tempo integral para lidar com apenas nosso canal.

Quando um quadro chega ao receptor, a soma de verificação é recalculada. Se o a soma de verificação no quadro está incorreta (ou seja, houve um erro de transmissão), os dados a camada de link é informada (*event = cksum err*). Se o quadro de entrada chegou intacta, a camada de enlace de dados também é informada (*evento = chegada de quadro*) para que pode adquirir o quadro para inspeção usando *da camada física*. Assim que o recebendo a camada de enlace de dados adquiriu um quadro não danificado, verifica o controle informações no cabeçalho e, se tudo estiver bem, passa a parte do pacote para a camada de rede. Sob nenhuma circunstância um cabeçalho de quadro é dado a um camada de rede.

Há uma boa razão pela qual a camada de rede nunca deve receber qualquer parte de o cabeçalho do quadro: para manter a rede e os protocolos de enlace de dados completamente separados

taxa. Contanto que a camada de rede não saiba absolutamente nada sobre o protocolo de enlace de dados

col ou o formato do quadro, essas coisas podem ser alteradas sem a necessidade de mudanças para o software da camada de rede. Isso acontece sempre que uma nova NIC é instalada em um computador. Fornecendo uma interface rígida entre as camadas de rede e de enlace de dados simplifica muito a tarefa de design porque os protocolos de comunicação em diferentes camadas podem evoluir de forma independente.

A Figura 3-11 mostra algumas declarações (em C) comuns a muitos dos protocolos para ser discutido mais tarde. Cinco estruturas de dados são definidas lá: *boolean*, *seq nr*, *pacote*, *tipo de quadro* e *quadro*. Um *booleano* é um tipo enumerado e pode assumir os valores são *verdadeiros* e *falsos*. Um *seq nr* é um pequeno inteiro usado para numerar os quadros para que possamos diferenciá-los. Esses números de sequência vão de 0 até e incluindo *MAX SEQ*, que é definido em cada protocolo que dele necessite. Um *pacote* é o unidade de informação trocada entre a camada de rede e a camada de enlace de dados na mesma máquina ou entre pares da camada de rede. Em nosso modelo sempre contém bytes *MAX PKT*, mas, de forma mais realista, teria comprimento variável. Um *frame* é composto de quatro campos: *kind*, *seq*, *ack* e *info*, os três primeiros dos que contêm informações de controle e a última das quais pode conter dados reais para ser transferido. Esses campos de controle são chamados coletivamente de **cabeçalho do quadro**. O campo *kind* informa se há algum dado no quadro, porque alguns dos os protocolos distinguem frames contendo apenas informações de controle daqueles contendo dados também. Os campos *seq* e *ack* são usados para números de sequência e agradecimentos, respectivamente; seu uso será descrito em mais detalhes mais tarde. O campo de *informações* de um quadro de dados contém um único pacote; o campo de *informação* de um

Página 243

SEC. 3,3

PROTOCOLOS DE LINK DE DADOS ELEMENTARES

219

quadro de controle não é usado. Uma implementação mais realista usaria uma variável campo de *informação de comprimento*, omitindo-o completamente para quadros de controle.

Novamente, é importante entender a relação entre um pacote e um quadro, Armação. A camada de rede constrói um pacote pegando uma mensagem do transporte camada e adicionando o cabeçalho da camada de rede a ela. Este pacote é passado para os dados camada de link para inclusão no campo de *informações* de um quadro de saída. Quando o quadro está

chega ao destino, a camada de enlace extrai o pacote do quadro e passa o pacote para a camada de rede. Desta forma, a camada de rede pode atuar como se as máquinas pudessesem trocar pacotes diretamente.

Vários procedimentos também estão listados na Figura 3-11. Estes são biblioteca detalhes cujos detalhes são dependentes da implementação e cujo funcionamento interno será não nos preocupa mais nas discussões seguintes. O procedimento *espera pelo evento* fica em um loop apertado esperando que algo aconteça, como mencionado anteriormente. o

procedimentos para a camada de rede e da camada de rede são usados pelo link de dados camada para passar pacotes para a camada de rede e aceitar pacotes da rede camada, respectivamente. Observe que da camada física e para a passagem da camada física quadros entre a camada de enlace de dados e a camada física. Em outras palavras, para camada de trabalho e da camada de rede lidar com a interface entre as camadas 2 e 3, enquanto da camada física e para a camada física lidam com a interface ser-entre as camadas 1 e 2.

Na maioria dos protocolos, presumimos que o canal não é confiável e perde quadros inteiros na ocasião. Para ser capaz de se recuperar de tais calamidades, o camada de enlace de envio de dados deve iniciar um cronômetro interno ou relógio sempre que enviar um

quadro, Armação. Se nenhuma resposta for recebida dentro de um certo intervalo de tempo predeterminado,

o relógio atinge o tempo limite e a camada de enlace de dados recebe um sinal de interrupção.

Em nossos protocolos isso é feito, permitindo que o processo *de espera para o evento de evento de retorno = tempo limite*. Os procedimentos *iniciam o cronômetro e param o cronômetro* giram o cronômetro

ligado e desligado, respectivamente. Os eventos de tempo limite são possíveis apenas quando o cronômetro é executado-

ning e antes que o temporizador de parada seja chamado. É explicitamente permitido chamar o *cronômetro de início*

enquanto o cronômetro está funcionando; tal chamada simplesmente zera o relógio para causar o próximo

tempo limite após um intervalo completo do cronômetro ter decorrido (a menos que seja redefinido ou desligado).

Os procedimentos *iniciam o cronômetro de confirmação e interrompem o controle do cronômetro auxiliar*.

usado para gerar reconhecimentos sob certas condições.

Os procedimentos de *habilitação da camada de rede e desabilitação da camada de rede* são usados em

os protocolos mais sofisticados, onde não assumimos mais que a rede camada sempre tem pacotes para enviar. Quando a camada de link de dados habilita a rede camada, a camada de rede tem permissão para interromper quando tem um pacote para ser enviei. Indicamos isso com *event = camada de rede pronta*. Quando a rede camada está desativada, pode não causar tais eventos. Tendo cuidado com quando ativa e desativa sua camada de rede, a camada de enlace de dados pode impedir a rede camada de inundá-lo com pacotes para os quais não tem espaço de buffer.

Os números de sequência do quadro estão sempre no intervalo de 0 a *MAX SEQ* (inclusive), onde *MAX SEQ* é diferente para os diferentes protocolos. É freqüentemente necessário

Página 244

220

A CAMADA DE LINK DE DADOS INDIVÍDUO. 3

para avançar um número de sequência em 1 circularmente (ou seja, *MAX SEQ* é seguido por 0).

A macro *inc* realiza este incremento. Foi definido como uma macro ser- porque ele é usado em linha dentro do caminho crítico. Como veremos mais tarde, o fator limitar o desempenho da rede é muitas vezes o processamento do protocolo, portanto, a definição de operações simples

erações como esta como macros não afetam a legibilidade do código, mas sim provar o desempenho.

As declarações da Figura 3-11 fazem parte de cada um dos protocolos que discutiremos Em breve. Para economizar espaço e fornecer uma referência conveniente, eles foram extraídos e listados juntos, mas conceitualmente eles devem ser mesclados com o protocolos próprios. Em C, essa fusão é feita colocando as definições em um arquivo de cabeçalho especial, neste caso *protocol.h*, e usando o recurso #include do C pré-processador para incluí-los nos arquivos de protocolo.

3.3.1 Um protocolo simplex utópico

Como um exemplo inicial, vamos considerar um protocolo que é tão simples quanto pode ser porque não se preocupa com a possibilidade de algo dar errado. Os dados são transmitido em apenas uma direção. Rede de transmissão e recepção as camadas estão sempre prontas. O tempo de processamento pode ser ignorado. O espaço infinito do buffer é

acessível. E o melhor de tudo, o canal de comunicação entre a camada de enlace de dados e a camada de rede nunca danifica ou perde quadros. Este protocolo totalmente irrealista, que vamos apelidar de "Utopia", é simplesmente para mostrar a estrutura básica na qual vou construir. Sua implementação é mostrada na Figura 3-12.

O protocolo consiste em dois procedimentos distintos, um emissor e um receptor. O remetente é executado na camada de enlace de dados da máquina de origem, e o receptor é executado em

a camada de enlace de dados da máquina de destino. Sem números de sequência ou reconhecimento de sequências, nem sequer é necessário. O único tipo de evento possível é a *chegada do quadro* (isto é, a chegada de um quadro não danificado).

O remetente está em um loop infinito enquanto apenas envia dados para a linha enquanto o mais rápido possível. O corpo do loop consiste em três ações: ir buscar um pacote da (sempre obrigada) camada de rede, construir um quadro de saída usando o variável *s*, e enviar o quadro em seu caminho. Somente a *informações* de campo do quadro é usada por este protocolo, porque os outros campos têm a ver com erro e controle de fluxo e não há erros ou restrições de controle de fluxo aqui.

O receptor é igualmente simples. Inicialmente, ele espera que algo aconteça, a única possibilidade é a chegada de um quadro não danificado. Eventualmente, o quadro chega e o procedimento *espera pelo retorno do evento*, com o evento definido para a *chegada do quadro*

(que é ignorado de qualquer maneira). A chamada para *da camada física* remove o novo chegou o quadro do buffer de hardware e o colocou na variável *r*, onde o receptor pode chegar lá. Finalmente, a parte dos dados é passada para a rede camada, e a camada de enlace de dados se acomoda para esperar pelo próximo quadro, efetivamente suspendendo-se até que o quadro chegue.

Página 245

SEC. 3,3

PROTOCOLOS DE LINK DE DADOS ELEMENTARES

221

```
/* Protocolo 1 (Utopia) fornece transmissão de dados em apenas uma direção, de
emissor para receptor. O canal de comunicação é considerado livre de erros
e o receptor é considerado capaz de processar todas as entradas infinitamente rapidamente.
Consequentemente, o remetente apenas fica em um loop, bombeando dados para a linha enquanto
o mais rápido possível. */
```

```
typedef enum {frame chegada} tipo de evento;
#include "protocol.h"
void sender1 (void)
{
frame s;
/* buffer para um quadro de saída */
buffer de pacote;
/* buffer para um pacote de saída */
enquanto (verdadeiro) {
da camada de rede (& buffer);
/* vá buscar algo para enviar */
s.info = buffer;
/* copie em s para transmissão */
para a (s) camada (s) física (s);
/* enviar em seu caminho */
}
/* Amanhã e amanhã e amanhã,
Arrasta-se neste ritmo mesquinho dia a dia
```

```

Até a última sílaba do tempo registrado.

- Macbeth, V, v * /
}

void receiver1 (void)
{
frame r;
evento de tipo de evento;
/* preenchido por esperar, mas não usado aqui */ 
enquanto (verdadeiro) {
esperar pelo evento (& evento);
/* A única possibilidade é a chegada de quadros */
da camada física (& r);
/* vá obter o quadro de entrada */
para a camada de rede (& r.info);
/* passa os dados para a camada de rede */
}
}

```

Figura 3-12. Um protocolo simplex utópico.

O protocolo da utopia não é realista porque não lida com nenhum fluxo de controle ou correção de erros. Seu processamento é próximo ao de uma confirmação não reconhecida serviço sem conexão que depende de camadas superiores para resolver esses problemas, embora mesmo um serviço sem conexão não confirmado faria alguma detecção de erro.

3.3.2 Um Protocolo Simplex Stop-and-Wait para um canal livre de erros

Agora vamos resolver o problema de evitar que o remetente inundar o receptor com quadros mais rápido do que o último é capaz de processá-los. Essa situação pode acontecer facilmente na prática, portanto, ser capaz de prevenir é de grande importância.

O canal de comunicação ainda é considerado livre de erros, no entanto, e o tráfego de dados ainda é simplex.

Uma solução é construir o receptor para ser poderoso o suficiente para processar uma constante fluxo contínuo de frames back-to-back (ou, de forma equivalente, defina a camada de link como lento o suficiente para que o receptor possa acompanhá-lo). Deve ter buffer suficiente e habilidades de processamento para funcionar na taxa de linha e deve ser capaz de passar os quadros que

são recebidos na camada de rede com rapidez suficiente. No entanto, este é o pior caso solução. Requer hardware dedicado e pode ser um desperdício de recursos se o uso da utilização do link é geralmente baixa. Além disso, isso apenas muda o problema de negociação com um remetente que é muito rápido em outro lugar; neste caso, para a camada de rede.

Uma solução mais geral para este problema é fazer com que o receptor forneça feedback de volta para o remetente. Depois de passar um pacote para sua camada de rede, o receptor envia um pequeno quadro fictício de volta para o remetente que, na verdade, dá ao remetente permissão para transmitir o próximo quadro. Depois de enviar um quadro, o remetente é re-exigido pelo protocolo para aguardar até que o pequeno manequim (ou seja, reconhecimento) chega o quadro. Esse atraso é um exemplo simples de protocolo de controle de fluxo. Protocolos nos quais o remetente envia um quadro e, em seguida, espera por um reconhecimento antes de prosseguir são chamados de **parar e esperar**. A Figura 3-13 mostra um exemplo de um protocolo simples de parar e esperar.

Embora o tráfego de dados neste exemplo seja simples, indo apenas do remetente para o receptor, os quadros viajam em ambas as direções. Consequentemente, a comunicação canal de conexão entre as duas camadas de enlace de dados deve ser capaz de bidirecional transferência de informação. No entanto, este protocolo envolve uma alternância estrita de fluxo: primeiro o remetente envia um quadro, então o receptor envia um quadro, então o remetente envia outro quadro, o receptor envia outro e assim por diante. Metade-canal físico duplex seria suficiente aqui.

Como no protocolo 1, o remetente começa buscando um pacote na rede

camada, usando-a para construir um quadro e enviando-o em seu caminho. Mas agora, ao contrário de

protocolo 1, o remetente deve esperar até que um quadro de confirmação chegue antes fazendo um loop de volta e buscando o próximo pacote da camada de rede. O envio camada de enlace de dados não precisa nem mesmo inspecionar o quadro de entrada, pois há apenas um sibilidade. O quadro de entrada é sempre uma confirmação.

A única diferença entre o *receptor 1* e o *receptor 2* é que depois de entregar um pacote para a camada de rede, o *receptor2* envia um quadro de confirmação de volta para o remetente antes de entrar no loop de espera novamente. Porque apenas a chegada do o quadro de volta para o remetente é importante, não seu conteúdo, o receptor não precisa colocar qualquer informação particular nele.

3.3.3 Um Protocolo Simplex Stop-and-Wait para um canal ruidoso

Agora, vamos considerar a situação normal de um canal de comunicação que comete erros. Os quadros podem ser danificados ou perdidos completamente. No entanto, nós suponha que se um quadro for danificado em trânsito, o hardware do receptor detectará isso

SEC. 3,3

PROTOCOLOS DE LINK DE DADOS ELEMENTARES

223

```
/* Protocolo 2 (Stop-and-Wait) também fornece um fluxo unidirecional de dados de
emissor para receptor. O canal de comunicação é mais uma vez considerado um erro
livre, como no protocolo 1. No entanto, desta vez, o receptor tem apenas um buffer finito
capacidade e uma velocidade de processamento finita, então o protocolo deve prevenir explicitamente
o remetente inundar o receptor com dados mais rápido do que pode ser manipulado. */  

typedef enum {frame chegada} tipo de evento;  

#include "protocol.h"  

void sender2 (void)  

{  

    frame s;  

    /* buffer para um quadro de saída */  

    buffer de pacote;  

    /* buffer para um pacote de saída */  

    evento de tipo de evento;  

    /* chegada de quadro é a única possibilidade */  

    enquanto (verdadeiro) {  

        da camada de rede (& buffer);  

        /* vá buscar algo para enviar */  

        s.info = buffer;  

        /* copie em s para transmissão */  

        para a (s) camada (s) física (s);  

        /* bye-bye little frame */  

        esperar pelo evento (& evento);  

        /* não prossiga até que seja dado o sinal verde */  

    }  

}  

void receiver2 (void)  

{  

    quadro r, s;  

    /* buffers para frames */  

    evento de tipo de evento;  

    /* chegada de quadro é a única possibilidade */  

    enquanto (verdadeiro) {  

        esperar pelo evento (& evento);  

        /* A única possibilidade é a chegada de quadros */  

        da camada física (& r);  

        /* vá obter o quadro de entrada */  

        para a camada de rede (& r.info);  

        /* passa os dados para a camada de rede */
```

```

para a (s) camada (s) física (s);
/* enviar um quadro fictício para despertar o remetente */
}
}

```

Figura 3-13. Um protocolo simples de parar e esperar.

quando ele calcula a soma de verificação. Se a estrutura estiver danificada de tal forma que o a soma de verificação está, no entanto, correta - uma ocorrência improvável - este protocolo (e todos outros protocolos) podem falhar (ou seja, entregar um pacote incorreto à camada de rede).

À primeira vista, pode parecer que uma variação do protocolo 2 funcionaria: adicionar um cronômetro. O remetente poderia enviar um quadro, mas o receptor enviaria apenas um ac- quadro de conhecimento se os dados foram recebidos corretamente. Se um quadro danificado estiver rived no receptor, seria descartado. Depois de um tempo, o remetente teria tempo e envie o quadro novamente. Este processo seria repetido até o quadro finalmente chegou intacto.

No entanto, este esquema tem uma falha fatal. Pense sobre o problema e tente descubra o que pode dar errado antes de continuar lendo.

224

A CAMADA DE LINK DE DADOS INDIVÍDUO. 3

Para ver o que pode dar errado, lembre-se de que o objetivo da camada de enlace é para fornecer comunicação transparente e livre de erros entre a camada de rede esses. A camada de rede na máquina *A* fornece uma série de pacotes para seu link de dados camada, que deve garantir que uma série idêntica de pacotes seja entregue à rede camada de trabalho na máquina *B* por sua camada de enlace. Em particular, a camada de rede em *B* não tem como saber se um pacote foi perdido ou duplicado, então o link de dados camada deve garantir que nenhuma combinação de erros de transmissão, embora ao contrário ly, pode fazer com que um pacote duplicado seja entregue a uma camada de rede.

Considere o seguinte cenário:

1. A camada de rede em *A* fornece o pacote 1 à sua camada de enlace de dados. o pacote é recebido corretamente em *B* e passado para a camada de rede em *B*. *B* envia um quadro de volta reconhecimento a *um*.
2. O quadro de confirmação se perde completamente. Simplesmente nunca é rives em tudo. A vida seria muito mais simples se o canal man- acumulou e perdeu apenas frames de dados e não frames de controle, mas é triste dizer, o canal não é muito discriminador.
3. A camada de enlace de dados em *A* eventualmente atinge o tempo limite. Não tendo recebido uma confirmação, ele (incorrectamente) assume que seu quadro de dados foi perdido ou danificado e envia o quadro contendo o pacote 1 novamente.
4. O quadro duplicado também chega intacto na camada de enlace de dados em *B* e é passado inadvertidamente para a camada de rede. Se *A* estiver enviando um arquivo para *B*, parte do arquivo será duplicada (ou seja, a cópia do arquivo feita por *B* estará incorreto e o erro não terá sido detectado). No outras palavras, o protocolo falhará.

Claramente, o que é necessário é alguma forma de o receptor ser capaz de distinguir um quadro que está vendo pela primeira vez em uma retransmissão. A maneira obvia para conseguir isso é fazer com que o remetente coloque um número de sequência no cabeçalho de cada quadro que ele envia. Em seguida, o receptor pode verificar o número de sequência de cada chegada quadro para ver se é um novo quadro ou uma duplicata a ser descartada.

Uma vez que o protocolo deve estar correto e o campo do número de sequência no cabeçalho mais provável que seja pequeno para usar o link de forma eficiente, surge a pergunta: qual é o número mínimo de bits necessários para o número de sequência? O cabeçalho pode pro vide 1 bit, alguns bits, 1 byte ou vários bytes para um número de sequência dependendo no protocolo. O ponto importante é que ele deve conter números de sequência que são grandes o suficiente para que o protocolo funcione corretamente ou não são exatamente um protocolo.

A única ambigüidade neste protocolo é entre um quadro, m , e seu sucessor direto cessador, $m + 1$. Se o quadro m for perdido ou danificado, o receptor não reconhecerá, portanto, o remetente continuará tentando enviá-lo. Uma vez recebido corretamente, o receptor enviará uma confirmação ao remetente. É aqui que o potencial

SEC. 3,3

PROTOCOLOS DE LINK DE DADOS ELEMENTARES

225

problemas surgem. Dependendo se o quadro de confirmação volta para o remetente corretamente ou não, o remetente pode tentar enviar m ou $m + 1$. No remetente, o evento que dispara a transmissão do quadro $m + 1$ é o arival de um reconhecimento para o quadro m . Mas esta situação implica que $m - 1$ foi recebido corretamente e, além disso, sua confirmação também foram recebidos corretamente pelo remetente. Caso contrário, o remetente não teria começado com m , muito menos considerando $m + 1$. Como consequência, o único ambigüidade é entre um quadro e seu predecessor ou sucessor imediato, não seja entre o predecessor e o sucessor.

Um número de sequência de 1 bit (0 ou 1) é, portanto, suficiente. A cada instante de vez, o receptor espera um determinado número de sequência a seguir. Quando um quadro con receber o número de sequência correto, é aceito e passado para a rede camada de trabalho, então reconhecida. Então, o número de sequência esperado é incremódulo 2 mentado (ou seja, 0 torna-se 1 e 1 torna-se 0). Qualquer quadro chegando con manter o número de sequência errado é rejeitado como uma duplicata. No entanto, o último confirmação válida é repetida para que o remetente possa eventualmente descobrir que o quadro foi recebido.

Um exemplo desse tipo de protocolo é mostrado na Figura 3-14. Protocolos nos quais o remetente espera por uma confirmação positiva antes de avançar para o próximo item de dados são frequentemente chamados de **ARQ** (solicitação de **repetição automática**) ou **PAR (positivo)**

Confirmação com retransmissão). Como o protocolo 2, este também transmits dados apenas em uma direção.

O protocolo 3 difere de seus predecessores porque tanto o remetente quanto o receptor têm um variável cujo valor é lembrado enquanto a camada de enlace de dados está no estado de espera. O remetente lembra o número de sequência do próximo quadro para enviar *próximo quadro a ser enviado*; o receptor lembra o número de sequência do próximo quadro esperado no *quadro esperado*. Cada protocolo tem uma curta fase de inicialização antes de entrar no loop infinito.

Depois de transmitir um quadro, o remetente inicia o cronômetro. Se fosse tudo pronto para funcionar, ele será reiniciado para permitir outro intervalo completo do cronômetro. O intervalo deve ser escolhido para permitir tempo suficiente para o quadro chegar ao receptor, para o receptor para processá-lo no pior caso, e para o quadro de confirmação propagar de volta para o remetente. Somente quando esse intervalo tiver decorrido é seguro para as suponha que o quadro transmitido ou sua confirmação foi perdido e para enviar uma cópia. Se o intervalo de tempo limite for definido muito curto, o remetente irá transmitir

quadros desnecessários. Embora esses quadros extras não afetem a exatidão do protocolo, eles vão prejudicar o desempenho.

Depois de transmitir um quadro e iniciar o cronômetro, o remetente espera por algum algo emocionante para acontecer. Existem apenas três possibilidades: um reconhecimento quadro chega sem danos, um quadro de confirmação danificado cambaleia, ou o temporizador expira. Se uma confirmação válida chegar, o remetente buscará o próximo pacote de sua camada de rede e o coloca no buffer, sobrescrevendo o anterior pacote. Também avança o número de sequência. Se um quadro danificado chegar ou o

226

A CAMADA DE LINK DE DADOS INDIVÍDUO. 3

temporizador expira, nem o buffer nem o número de sequência são alterados para que um duplicado pode ser enviado. Em todos os casos, o conteúdo do buffer (seja o próximo pacote et ou uma duplicata) são então enviados.

Quando um quadro válido chega ao receptor, seu número de sequência é verificado para veja se é uma duplicata. Caso contrário, é aceito, passado para a camada de rede e um confirmação é gerada. Duplicatas e frames danificados não são passados para a camada de rede, mas fazem com que o último quadro recebido corretamente seja confirmado para sinalizar ao remetente para avançar para o próximo quadro ou retransmitir um quadro danificado.

3.4 PROTOCOLOS DE JANELA DE DESLIZAMENTO

Nos protocolos anteriores, os quadros de dados eram transmitidos apenas em uma direção.

Na maioria das situações práticas, é necessário transmitir dados em ambas as direções.

Uma maneira de obter transmissão de dados full-duplex é executar duas instâncias de um dos protocolos anteriores, cada um usando um link separado para tráfego de dados simplex (em direções diferentes). Cada link é então composto por um canal de "encaminhamento" (para dados) e um canal "reverso" (para confirmações). Em ambos os casos, a capacidade o canal reverso é quase totalmente desperdiçado.

Uma ideia melhor é usar o mesmo link para dados em ambas as direções. Afinal, em protocolos 2 e 3 já estava sendo usado para transmitir frames nos dois sentidos, e o O canal reverso normalmente tem a mesma capacidade do canal direto. Nisso modelar os quadros de dados de *A* a *B* são misturados com o reconhecimento quadros de *um* a *B*. Olhando para o campo de *tipo* no cabeçalho de uma entrada quadro, o receptor pode dizer se o quadro é de dados ou uma confirmação.

Embora intercalar dados e quadros de controle no mesmo link seja um grande problema

prova sobre ter dois links físicos separados, outra melhoria é

possível. Quando um quadro de dados chega, em vez de enviar imediatamente um quadro de controle, o receptor se restringe e espera até que a camada de rede passe é o próximo pacote. A confirmação é anexada ao quadro de dados de saída

(usando o campo *ack* no cabeçalho do quadro). Na verdade, o reconhecimento obtém um carona gratuita no próximo quadro de dados de saída. A técnica de atrasar temporariamente confirmações de saída para que possam ser conectadas à próxima mensagem de saída o frame de dados é conhecido como **piggybacking**.

A principal vantagem de usar piggybacking sobre ter reconhecimento distinto edgement frames é um melhor uso da largura de banda do canal disponível. O campo *ack* no cabeçalho do quadro custa apenas alguns bits, enquanto um quadro separado precisaria de um cabeçalho, a confirmação e uma soma de verificação. Além disso, menos frames enviados geralmente significa uma carga de processamento mais leve no receptor. No próximo protocolo para ser examinado, o campo piggyback custa apenas 1 bit no cabeçalho do quadro. Raramente custa mais do que alguns bits.

No entanto, o piggybacking introduz uma complicação não presente com Reconhecimentos. Quanto tempo deve a camada de enlace de dados esperar por um pacote para

SEC. 3,4

PROTOCOLOS DE JANELA DE DESLIZAMENTO

227

```
/* O protocolo 3 (PAR) permite o fluxo de dados unidirecional em um canal não confiável. */
```

```
# define MAX SEQ 1
```

```
/* deve ser 1 para o protocolo 3 */
```

```
typedef enum {chegada de quadro, cksum err, tempo limite} tipo de evento;
```

```
#include "protocol.h"
```

```
void sender3 (void)
```

```
{
```

```

seq nr próximo quadro a ser enviado;
/* número seq do próximo quadro de saída */
frame s;
/* variável de rascunho */
buffer de pacote;
/* buffer para um pacote de saída */
evento de tipo de evento;
próximo quadro a ser enviado = 0;
/* inicializar números de sequência de saída */
da camada de rede (& buffer);
/* buscar o primeiro pacote */
enquanto (verdadeiro) {
    s.info = buffer;
    /* construir um quadro para transmissão */
    s.seq = próximo quadro a ser enviado;
    /* inserir número de sequência no quadro */
    para a (s) camada (s) física (s);
    /* enviar em seu caminho */
    iniciar cronômetro (s.seq);
    /* se a resposta demorar muito, tempo limite */
    esperar pelo evento (& evento);
    /* chegada de quadro, erro cksum, tempo limite */
    if (evento == chegada de quadro) {
        da camada física (& s);
        /* obter o reconhecimento */
        if (s.ack == próximo quadro a enviar) {
            parar o cronômetro (s.ack);
            /* desligar o cronômetro */
            da camada de rede (& buffer); /* obter o próximo a enviar */
            inc (próximo quadro a ser enviado);
            /* inverter o próximo quadro para enviar */
        }
    }
}
void receiver3 (void)
{
seq nr quadro esperado;
quadro r, s;
evento de tipo de evento;
quadro esperado = 0;
enquanto (verdadeiro) {
esperar pelo evento (& evento);
/* possibilidades: chegada de quadro, cksum err */
if (evento == chegada de quadro) {
/* um quadro válido chegou */
da camada física (& r);
/* vá buscar o quadro recém-chegado */
if (r.seq == quadro esperado) {
/* isso é o que estávamos esperando */
para a camada de rede (& r.info);
/* passa os dados para a camada de rede */
inc (quadro esperado);
/* da próxima vez, espere a outra sequência nr */
}
s.ack = 1 - quadro esperado;
/* dizer qual frame está sendo confirmado */
para a (s) camada (s) física (s);
/* enviar confirmação */
}
}
}

```

Figura 3-14. Um reconhecimento positivo com protocolo de retransmissão.

228

A CAMADA DE LINK DE DADOS

INDIVÍDUO. 3

qual pegar carona no reconhecimento? Se a camada de link de dados esperar mais do que o tempo limite do remetente, o quadro será retransmitido, derrotando o propósito inteiro de ter agradecimentos. Se a camada de enlace fosse um oráculo e poderia prever o futuro, saberia quando o próximo pacote da camada de rede iria entrar e poderia decidir esperar por ele ou enviar um ac-

conhecimento imediato, dependendo de quanto tempo foi a espera projetada

será. Claro, a camada de enlace de dados não pode prever o futuro, por isso deve recorrer a algum esquema ad hoc, como esperar um número fixo de milissegundos. E se um novo pacote chega rapidamente, a confirmação é adicionada a ele.

Caso contrário, se nenhum novo pacote chegar ao final deste período de tempo, os dados a camada de link apenas envia um quadro de confirmação separado.

Os próximos três protocolos são protocolos bidirecionais que pertencem a uma classe chamada protocolos de **janela deslizante de led**. Os três diferem entre si em termos de ef-

requisitos de eficiência, complexidade e buffer, conforme discutido posteriormente. Nestes, como

em

todos os protocolos de janela deslizante, cada quadro de saída contém um número de sequência, variando de 0 até algum máximo. O máximo é geralmente $2^n - 1$ então o se-

o número de sequência se encaixa exatamente em um campo de n bits. A janela deslizante pare e espere

protocolo usa $n = 1$, restringindo os números de sequência a 0 e 1, mas mais sofis

versões indicadas podem usar um n arbitrário .

A essência de todos os protocolos de janela deslizante é que, a qualquer momento, o

o remetente mantém um conjunto de números de sequência correspondentes aos quadros que é

permitido-

ted para enviar. Diz-se que esses quadros estão dentro da **janela de envio** . Similarmente,

o receptor também mantém uma **janela de recepção** correspondente ao conjunto de quadros

é permitido aceitar. A janela do remetente e a janela do receptor precisam

não ter os mesmos limites inferior e superior ou mesmo ter o mesmo tamanho. Em alguns

protocolos eles são fixos em tamanho, mas em outros eles podem aumentar ou diminuir em relação

ao

ao longo do tempo, à medida que os frames são enviados e recebidos.

Embora esses protocolos dêem à camada de enlace de dados mais liberdade sobre o

ordem na qual ele pode enviar e receber frames, definitivamente não descartamos o

requisito de que o protocolo deve entregar pacotes para a camada de rede de destino

na mesma ordem, eles foram passados para a camada de enlace de dados na máquina de envio.

Nem mudamos o requisito de que o canal de comunicação físico seja

" semelhante a um fio ", ou seja, deve entregar todos os frames do pedido enviado.

Os números de sequência na janela do remetente representam frames que têm

foram enviados ou podem ser enviados, mas ainda não foram confirmados. Sempre que um novo

pacote

chega da camada de rede, recebe o próximo número de sequência mais alto, e

a borda superior da janela é avançada em um. Quando um reconhecimento

entra, a borda inferior é avançada em um. Desta forma, a janela continua

ously mantém uma lista de frames não confirmados. A Figura 3-15 mostra um exemplo.

Uma vez que os frames atualmente na janela do remetente podem ser perdidos ou

danificados em trânsito, o remetente deve manter todos esses quadros em sua memória por

possível retransmissão. Assim, se o tamanho máximo da janela for n , o remetente precisa

n buffers para armazenar os quadros não confirmados. Se a janela crescer para o seu

Remetente

Receptor

7

6

1

5

2

0

4

3

7

6

1

5

2

0

4

3

7

6

1

5

2

0

4

3

7

6

1

5

2

0

4

3

7

6

1

5

2

0

4

3

7

6

1

5

2

0

4

3

7

6

1

5

2

0

4

3

(uma)

(b)

(c)

(d)

Figura 3-15. Uma janela deslizante de tamanho 1, com um número de sequência de 3 bits. (a) Inicialmente. (b) Depois que o primeiro quadro foi enviado. (c) Depois que o primeiro quadro foi recebido. (d) Após o primeiro aviso ter sido recebido.

tamanho máximo, a camada de enlace de dados de envio deve desligar a rede à força camada até que outro buffer fique livre.

A janela da camada de enlace de dados de recebimento corresponde aos quadros que ela pode acessar

exceto Qualquer quadro que caia na janela é colocado no buffer do receptor. Quando um quadro cujo número de sequência é igual à borda inferior da janela é re-

recebido, ele é passado para a camada de rede e a janela é girada em um. Qualquer quadro que caia para fora da janela é descartado. Em todos esses casos, um subsequente a confirmação é gerada para que o remetente possa descobrir como proceder.

Observe que um tamanho de janela de 1 significa que a camada de enlace só aceita quadros em ordem, mas para janelas maiores não é assim. A camada de rede, em contraste, é tudo maneiras alimentam os dados na ordem adequada, independentemente do tamanho da janela da camada de link de dados.

A Figura 3-15 mostra um exemplo com um tamanho máximo de janela de 1. Inicialmente, nenhum quadro é destacado, então as bordas inferior e superior da janela do remetente são iguais, mas conforme o tempo passa, a situação progride conforme mostrado. Ao contrário do send-

janela er, a janela do receptor sempre permanece em seu tamanho inicial, girando como o próximo quadro é aceito e entregue à camada de rede.

3.4.1 Um protocolo de janela deslizante de um bit

Antes de abordar o caso geral, vamos examinar um protocolo de janela deslizante com um tamanho de janela de 1. Esse protocolo usa parar e esperar, pois o remetente transmite um quadro e aguarda sua confirmação antes de enviar o próximo.

Página 254

230

A CAMADA DE LINK DE DADOS INDIVÍDUO. 3

A Figura 3-16 descreve esse protocolo. Como os outros, começa definindo algumas variáveis. *O próximo quadro a ser enviado* informa qual quadro o remetente está tentando enviar. Da mesma forma, o *quadro esperado* informa qual quadro o receptor está esperando. No ambos os casos, 0 e 1 são as únicas possibilidades.

```
/* O protocolo 4 (janela deslizante) é bidirecional. */
#define MAX SEQ 1
/* deve ser 1 para o protocolo 4 */
typedef enum {chegada de quadro, cksum err, tempo limite} tipo de evento;
#include "protocol.h"
void protocol4 (void)
{
    seq nr próximo quadro a ser enviado;
    /* 0 ou 1 apenas */
    seq nr quadro esperado;
    /* 0 ou 1 apenas */
    quadro r, s;
    /* variáveis de rascunho */
    buffer de pacote;
    /* pacote atual sendo enviado */
    evento de tipo de evento;
    próximo quadro a ser enviado = 0;
    /* próximo quadro no fluxo de saída */
    quadro esperado = 0;
    /* quadro esperado próximo */
    da camada de rede (& buffer);
    /* buscar um pacote da camada de rede */
    s.info = buffer;
    /* prepare-se para enviar o quadro inicial */
    s.seq = próximo quadro a ser enviado;
    /* inserir número de sequência no quadro */
    s.ack = 1 - quadro esperado;
    /* confirmação agregada */
    para a (s) camada (s) física (s);
    /* transmitir o quadro */
    iniciar cronômetro (s.seq);
    /* inicia a execução do cronômetro */
    enquanto (verdadeiro) {
```

```

esperar pelo evento (& evento);
/* chegada de quadro, erro cksum ou tempo limite */
if (evento == chegada de quadro) {
/* um quadro chegou sem danos */
da camada física (& r);
/* vá buscar */
if (r.seq == quadro esperado) {
/* lidar com fluxo de quadro de entrada */
para a camada de rede (& r.info);
/* passar o pacote para a camada de rede */
inc (quadro esperado);
/* invert seq number esperado próximo */
}
if (r.ack == próximo quadro a enviar) {
/* lidar com fluxo de quadro de saída */
parar o cronômetro (r.ack);
/* desligar o cronômetro */
da camada de rede (& buffer);
/* buscar novo pacote da camada de rede */
inc (próximo quadro a ser enviado);
/* inverter o número de sequência do remetente */
}
}
s.info = buffer;
/* construir quadro de saída */
s.seq = próximo quadro a ser enviado;
/* inserir número de sequência nele */
s.ack = 1 - quadro esperado;
/* número de sequência do último quadro recebido */
para a (s) camada (s) física (s);
/* transmitir um quadro */
iniciar cronômetro (s.seq);
/* inicia a execução do cronômetro */
}
}

```

Figura 3-16. Um protocolo de janela deslizante de 1 bit.

Página 255

SEC. 3,4

PROTÓCOLOS DE JANELA DE DESLIZAMENTO

231

Em circunstâncias normais, uma das duas camadas de enlace de dados vai primeiro e transmite o primeiro quadro. Em outras palavras, apenas um dos programas da camada de enlace de dados

deve conter a *camada física* e *iniciar* chamadas de procedimento de *cronômetro* fora do loop principal. A máquina inicial busca o primeiro pacote de sua camada de rede, constrói um quadro a partir dele e o envia. Quando este (ou qualquer) quadro chega, o receptor A camada de enlace de dados verifica se é uma duplicata, assim como no protocolo 3. Se o quadro é o esperado, ele é passado para a camada de rede e o receptor dow é deslizado para cima.

O campo de confirmação contém o número do último quadro recebido sem erro. Se este número estiver de acordo com o número de sequência do quadro, o remetente está tentando enviar, o remetente sabe que foi feito com o quadro armazenado em *buffer* e pode buscar o próximo pacote de sua camada de rede. Se o número da sequência discorda, deve continuar tentando enviar o mesmo quadro. Sempre que um quadro é recebido, um quadro também é enviado de volta.

Agora, vamos examinar o protocolo 4 para ver como ele é resiliente a cenários patológicos arios. Suponha que o computador A está tentando enviar seu quadro 0 para o computador B e que B está tentando enviar seu quadro 0 a Um . Suponha que A envie um quadro para B , mas

O intervalo de tempo limite de A é um pouco curto. Consequentemente, A pode expirar repetidamente,

enviando uma série de quadros idênticos, todos com $seq = 0$ e $ack = 1$.

Quando o primeiro frame válido chegar ao computador B, ele será aceito e *quadro esperado* será definido com um valor de 1. Todos os quadros subsequentes recebidos será rejeitado porque B agora está esperando quadros com número de sequência 1, não 0. Além disso, uma vez que todas as duplicatas terão $ack = 1$ e B ainda está esperando por uma confirmação de 0, B não irá buscar um novo pacote de sua rede camada.

Depois que cada duplicata rejeitada chegar, B enviará a A um quadro contendo $seq = 0$ e $ack = 0$. Eventualmente, um deles chegará corretamente em A, causando A para começar a enviar o próximo pacote. Nenhuma combinação de frames perdidos ou prematuros os tempos limite podem fazer com que o protocolo entregue pacotes duplicados para qualquer rede camada, para pular um pacote ou para um deadlock. O protocolo está correto.

No entanto, para mostrar como as interações de protocolo sutis podem ser, notamos que um pe-situação culiar surge se ambos os lados enviam simultaneamente um pacote inicial. Isto A dificuldade de sincronização é ilustrada pela Figura 3-17. Na parte (a), a operação normal do protocolo é mostrado. Em (b) a peculiaridade é ilustrada. Se B esperar por O primeiro quadro de A antes de enviar um seu próprio, a sequência é conforme mostrado em (a), e cada quadro é aceito.

No entanto, se A e B iniciarem a comunicação simultaneamente, seus primeiros quadros cruz e as camadas de enlace de dados entram na situação (b). Em (a) cada chegada de quadro traz um novo pacote para a camada de rede; não há duplicatas. Em (b) metade de os quadros contêm duplicatas, embora não haja erros de transmissão. Simi-situações lar podem ocorrer como resultado de timeouts prematuros, mesmo quando um lado claramente começa primeiro. Na verdade, se ocorrerem vários tempos limite prematuros, os quadros podem ser enviado três ou mais vezes, desperdiçando largura de banda valiosa.

232

A CAMADA DE LINK DE DADOS

INDIVÍDUO. 3

A envia (0, 1, A0)
A obtém (0, 0, B0) *
A envia (1, 0, A1)
B obtém (0, 1, A0) *
B envia (0, 0, B0)
B obtém (1, 0, A1) *
B envia (1, 1, B1)
B obtém (0, 1, A2) *
B envia (0, 0, B2)
B obtém (1, 0, A3) *
B envia (1, 1, B3)
A obtém (1, 1, B1) *
A envia (0, 1, A2)
A obtém (0, 0, B2) *
A envia (1, 0, A3)
A envia (0, 1, A0)
A obtém (0, 1, B0) *
A envia (0, 0, A0)
B obtém (0, 0, A0)
B envia (1, 0, B1)
B envia (0, 1, B0)
B obtém (0, 1, A0) *
B envia (0, 0, B0)
B obtém (1, 0, A1) *
B envia (1, 1, B1)
B obtém (1, 1, A1)
B envia (0, 1, B2)
A obtém (0, 0, B0)
A envia (1, 0, A1)
A obtém (1, 0, B1) *
A envia (1, 1, A1)
Tempo
(uma)
(b)

Figura 3-17. Dois cenários para o protocolo 4. (a) Caso normal. (b) Anormal caso. A notação é (seq , ack , número do pacote). Um asterisco indica onde um

camada de rede aceita um pacote.

3.4.2 Um protocolo usando Go-Back-N

Até agora, fizemos a suposição tácita de que o tempo de transmissão necessário para que um quadro chegue ao receptor mais o tempo de transmissão para o acionamento do reconhecimento para voltar é insignificante. Às vezes, essa suposição é claramente falso. Nessas situações, o longo tempo de ida e volta pode ter implicações importantes para a eficiência da utilização da largura de banda. Por exemplo, considere um 50 kbps canal de satélite com um atraso de propagação de ida e volta de 500 ms. Vamos imaginar tentando usar o protocolo 4 para enviar quadros de 1000 bits via satélite. Em $t = 0$ o remetente começa a enviar o primeiro quadro. Em $t = 20$ mseg, o quadro foi concluído completamente enviado. Não até $t = 270$ mseg é que o quadro chega totalmente ao receptor, e não até $t = 520$ mseg é que o reconhecimento chegou de volta ao remetente, na melhor das circunstâncias (sem espera no receptor e um curto atraso de reconhecimento). Isso significa que o remetente foi bloqueado 500/520 ou 96% do tempo. Em outras palavras, apenas 4% da largura de banda disponível foi utilizada. Claro-Sim, a combinação de um longo tempo de trânsito, alta largura de banda e comprimento de quadro curto é desastroso em termos de eficiência.

O problema descrito aqui pode ser visto como uma consequência da regra exigindo que um remetente espere por uma confirmação antes de enviar outro quadro. E se relaxando essa restrição, uma eficiência muito melhor pode ser alcançada. Basicamente, a solução está em permitir que o remetente transmita até w quadros antes de bloquear, em vez de apenas 1. Com uma escolha grande o suficiente de w , o remetente será capaz de transmitir quadros continuamente, já que as confirmações chegarão para quadros antes de a janela ficar cheia, evitando que o remetente bloqueie.

Página 257

SEC. 3,4

PROTOCOLOS DE JANELA DE DESLIZAMENTO

233

Para encontrar um valor apropriado para w , precisamos saber quantos quadros podem caber dentro do canal à medida que se propagam do emissor para o receptor. Esta capacidade é determinada pela largura de banda em bits / s multiplicada pelo tempo de trânsito unidirecional, ou o **produto de atraso de largura de banda** do link. Podemos dividir essa quantidade pelo número de bits em um quadro para expressá-lo como um número de quadros. Chame esta quantidade BD . Então w deve ser definido como $2 BD + 1$. Duas vezes o atraso da largura de banda é o número de quadros que podem ser pendentes se o remetente enviar quadros continuamente quando o tempo de ida e volta para receber uma confirmação é considerado. O "+1" é porque um quadro de confirmação não será enviado até depois de um quadro completo é recebido.

Para o exemplo de link com uma largura de banda de 50 kbps e um tempo de trânsito unidirecional de 250 ms, o produto de atraso de largura de banda é de 12,5 kbit ou 12,5 quadros de 1000 bits cada. $2 BD + 1$ tem então 26 quadros. Suponha que o remetente comece a enviar o quadro 0 como antes e envia um novo quadro a cada 20 ms. Quando terminar de enviar 26 quadros, em $t = 520$ mseg, o reconhecimento para o quadro 0 terá apenas chegado ao receptor. Depois disso, as confirmações chegarão a cada 20 ms, então o remetente sempre obterá permissão para continuar apenas quando for necessário. A partir de então, 25 ou 26 frames não reconhecidos sempre estarão pendentes. Em outros termos, o tamanho máximo da janela do remetente é 26.

Para tamanhos de janela menores, a utilização do link será inferior a 100% uma vez que o remetente será bloqueado às vezes. Podemos escrever a utilização como a fração de tempo em que o remetente não está bloqueado:

utilização do link \leq

$$1 + 2 BD$$

W

Este valor é um limite superior porque não permite o processamento de nenhum quadro no tempo e trata o quadro de reconhecimento como tendo comprimento zero, uma vez que é

geralmente curto. As equação mostra a necessidade de ter uma grande janela w quando sempre o produto de atraso de largura de banda é grande. Se o atraso for alto, o remetente irá esgotar sua janela mesmo para uma largura de banda moderada, como no satélite exemplo. Se a largura de banda for alta, mesmo por um atraso moderado, o remetente irá esgotar sua janela rapidamente, a menos que tenha uma grande janela (por exemplo, um link de 1 Gbps com

Atraso de 1 ms contém 1 megabit). Com parar e esperar para o qual $w = 1$, se houver mesmo o valor de um frame de atraso de propagação, a eficiência será inferior a 50%.

Esta técnica de manter vários quadros em vôo é um exemplo de **pipelining**. O pipelining de frames em um canal de comunicação não confiável levanta algumas questões sérias. Primeiro, o que acontece se um quadro no meio de um longo fluxo é danificado ou perdido? Um grande número de quadros sucessivos chegará ao receptor antes mesmo que o remetente descubra que algo está errado. Quando um quadro danificado chega ao receptor, obviamente deve ser descartado, mas o que deveria receber fazer com todos os frames corretos que o seguem? Lembre-se que o recebimento A camada de enlace de dados é obrigada a entregar os pacotes à camada de rede em sequência.

Página 258

234

A CAMADA DE LINK DE DADOS INDIVÍDUO. 3

Duas abordagens básicas estão disponíveis para lidar com erros na presença de pipelining, ambos mostrados na Fig. 3-18.

0
1
0
1
2
3
4
5
6
7
8
E
D
D
D
D
D
2
3
4
5
6
7
8
2
3
4
5
6
7
8
9
Intervalo de tempo limite
Erro
Frames descartados pela camada de enlace de dados
Quadros armazenados em buffer
por camada de link de dados
Ack0
Ack1
Tempo
(uma)
(b)
0
1
0
1
9
10
11
12

```
13
14
E
2
3
4
5
2
6
7
8
9
10
11
12
13
14
15
8
Erro
Ack
0
Ack
1
Nak
2
4
5
2
3
6
Ack
5
Ack
6
7
Ack
7
Ack
8
Ack
9
Ack
11
Ack
12
Ack
13
Ack
10
Ack
2
Ack
3
Ack
4
Ack
5
Ack
6
Ack
7
```

Figura 3-18. Pipelining e recuperação de erros. Efeito de um erro quando

(a) o tamanho da janela do receptor é 1 e (b) o tamanho da janela do receptor é grande.

Uma opção, chamada **go-back-n**, é para o receptor simplesmente descartar todos os frames, sem enviar reconhecimentos para os frames descartados. Esta estratégia corresponde a uma janela de recepção de tamanho 1. Em outras palavras, a camada de enlace de dados

recusa-se a aceitar qualquer quadro, exceto o próximo, que deve ser fornecido à camada de rede. Se a janela do remetente ficar cheia antes que o tempo acabe, o pipeline começará a vazio. Eventualmente, o remetente atingirá o tempo limite e retransmitirá todos os não confirmados quadros em ordem, começando com o danificado ou perdido. Esta abordagem pode desperdiçar um muita largura de banda se a taxa de erro for alta.

Na Fig. 3-18 (b), vemos go-back-n para o caso em que a janela do receptor é grande. Os quadros 0 e 1 são recebidos e confirmados corretamente. Quadro 2, no entanto, está danificado ou perdido. O remetente, sem saber desse problema, continua a enviar quadros até que o cronômetro para o quadro 2 expire. Em seguida, ele volta para o frame 2 e recomeça com ele, enviando 2, 3, 4, etc. novamente.

A outra estratégia geral para lidar com erros quando os frames são canalizados é

chamada de **repetição seletiva**. Quando é usado, um quadro ruim recebido é descartado, mas todos os quadros bons recebidos após são aceitos e armazenados em buffer. Quando o remetente expira, apenas o quadro não confirmado mais antigo é retransmitido. Se aquele quadro

Página 259

SEC. 3,4

PROTOCOLOS DE JANELA DE DESLIZAMENTO

235

chega corretamente, o receptor pode entregar para a camada de rede, em sequência, todos os quadros que armazenou em buffer. A repetição seletiva corresponde a uma janela do receptor maior do que 1. Esta abordagem pode exigir grandes quantidades de memória da camada de enlace se o a janela é grande.

A repetição seletiva é frequentemente combinada com o envio de uma negativa pelo receptor confirmação (NAK) quando detecta um erro, por exemplo, quando recebe um erro de checksum ou quadro fora de sequência. NAKs estimulam a retransmissão antes que o temporizador correspondente expire e, assim, melhore o desempenho.

Na Fig. 3-18 (b), os quadros 0 e 1 são novamente recebidos e confirmados corretamente e o quadro 2 é perdido. Quando o quadro 3 chega ao receptor, a camada de enlace de dados percebe que perdeu um quadro e, portanto, envia de volta um NAK para 2, mas armazena 3.

Quando os quadros 4 e 5 chegam, eles também são armazenados em buffer pela camada de enlace de dados.

de ser passado para a camada de rede. Eventualmente, o NAK 2 volta ao remetente, que reenvia imediatamente o quadro 2. Quando ele chega, a camada de enlace de dados agora tem 2, 3, 4 e 5 e pode passar todos eles para a camada de rede no ordem. Ele também pode reconhecer todos os frames até e incluindo 5, conforme mostrado no figura. Se o NAK se perder, eventualmente o remetente atingirá o tempo limite para o quadro 2 e enviá-lo (e apenas ele) por conta própria, mas isso pode demorar um pouco mais tarde.

Essas duas abordagens alternativas são compensações entre o uso eficiente de banda largura e espaço de buffer da camada de enlace. Dependendo de qual recurso é mais escasso, um ou outro pode ser usado. A Figura 3-19 mostra um protocolo go-back-n no qual a camada de enlace de dados de recebimento aceita apenas quadros em ordem; quadros seguindo um erros são descartados. Neste protocolo, pela primeira vez, eliminamos o as-

supondo que a camada de rede sempre tenha um suprimento infinito de pacotes para enviar.

Quando a camada de rede tem um pacote que deseja enviar, pode causar uma *rede camada de trabalho pronta* para o evento acontecer. No entanto, para fazer cumprir o limite de controle de fluxo em

a janela do remetente ou o número de frames não confirmados que podem estar fora permanente a qualquer momento, a camada de enlace de dados deve ser capaz de manter a camada de rede

de incomodá-lo com mais trabalho. Os procedimentos da biblioteca *permitem a camada de rede e desativar a camada de rede para* fazer este trabalho.

O número máximo de frames que podem estar pendentes a qualquer momento não é o mesmo que o tamanho do espaço do número de sequência. Para voltar-n, *MAX SEQ* os quadros podem estar pendentes a qualquer momento, mesmo que haja *MAX SEQ + 1* números de sequência distintos (que são 0, 1, ..., *MAX SEQ*). Veremos um restrição ainda mais rígida para o próximo protocolo, repetição seletiva. Para ver por que isso res- a tração é necessária, considere o seguinte cenário com *MAX SEQ = 7*:

1. O remetente envia os quadros de 0 a 7.
2. Uma confirmação associada a 7 volta para o remetente.
3. O remetente envia outros oito quadros, novamente com números de sequência 0 a 7.
4. Agora, outro reconhecimento associado ao quadro 7 chega.

Página 260

236

A CAMADA DE LINK DE DADOS

INDIVÍDUO. 3

```
/* O protocolo 5 (Go-back-n) permite vários quadros pendentes. O remetente pode transmitir para MAX SEQ frames sem esperar por um ack. Além disso, ao contrário do anterior protocolos, a camada de rede não deve ter um novo pacote o tempo todo. Em vez de, a camada de rede causa um evento de pronta para a camada de rede quando há um pacote a ser enviado. */  
# define MAX SEQ 7  
typedef enum {chegada de quadro, cksum err, tempo limite, camada de rede pronta} tipo de evento;  
#include "protocol.h"  
booleano estático entre (seq nr a, seq nr b, seq nr c)  
{  
    /* Retorna verdadeiro se a <= b <c circularmente; caso contrário, false. */  
    if (((a <= b) && (b <c)) || ((c <a) && (a <= b)) || ((b <c) && (c <a)))  
        return (true);  
    outro  
    retorna falso;  
}  
estático void enviar dados (seq nr frame nr, seq nr frame esperado, buffer de pacote [])  
{  
    /* Construir e enviar um quadro de dados. */  
    frame s;  
    /* variável de rascunho */  
    s.info = buffer [nr frame];  
    /* inserir pacote no quadro */  
    s.seq = frame nr;  
    /* inserir número de sequência no quadro */  
    s.ack = (frame esperado + MAX SEQ)% (MAX SEQ + 1); /* piggyback ack */  
    para a (s) camada (s) física (s);  
    /* transmitir o quadro */  
    iniciar o temporizador (nº do quadro);  
    /* inicia a execução do cronômetro */  
}  
void protocol5 (void)  
{  
    seq nr próximo quadro a ser enviado;  
    /* SEQ MÁX.> 1; usado para fluxo de saída */  
    seq nr ack esperado;  
    /* quadro mais antigo ainda não reconhecido */  
    seq nr quadro esperado;  
    /* próximo quadro esperado no fluxo de entrada */  
    frame r;  
    /* variável de rascunho */  
    buffer de pacote [MAX SEQ + 1];  
    /* buffers para o fluxo de saída */  
    seq nr nbuffed;  
    /* número de buffers de saída atualmente em uso */  
    seq nr i;  
    /* usado para indexar na matriz de buffer */  
    evento de tipo de evento;  
    habilitar camada de rede ();  
    /* permitir eventos prontos da camada de rede */  
    ack esperado = 0;  
    /* próxima confirmação esperada de entrada */  
    próximo quadro a ser enviado = 0;  
    /* próximo quadro saindo */  
    quadro esperado = 0;  
    /* número de quadro esperado de entrada */  
    nbuffed = 0;  
    /* inicialmente nenhum pacote é armazenado em buffer */  
    enquanto (verdadeiro) {  
        esperar pelo evento (& evento);  
        /* quatro possibilidades: ver tipo de evento acima */  
    }
```

SEC. 3,4
PROTOCOLOS DE JANELA DE DESLIZAMENTO

237

```

switch (evento) {
    camada de rede do caso pronta:
        /* a camada de rede tem um pacote para enviar */
        /* Aceitar, salvar e transmitir um novo quadro. */
        da camada de rede (& buffer [próximo quadro a enviar]); /* buscar novo pacote */
        nbuffed = nbuffed + 1;
        /* expandir a janela do remetente */
        enviar dados (próximo quadro a enviar, quadro esperado, buffer); /* transmitir o quadro */
        inc (próximo quadro a ser enviado);
        /* avançar a borda superior da janela do remetente */
        quebrar;
        chegada da moldura da caixa:
            /* um dado ou quadro de controle chegou */
            da camada física (& r);
            /* obter quadro de entrada da camada física */
            if (r.seq == quadro esperado) {
                /* Quadros são aceitos apenas em ordem. */
                para a camada de rede (& r.info);
                /* passar o pacote para a camada de rede */
                inc (quadro esperado);
                /* avançar borda inferior da janela do receptor */
            }
            /* Confirmação n implica n - 1, n - 2, etc. Verifique isso. */
            while (entre (ack esperado, r.ack, próximo quadro a ser enviado)) {
                /* Lidar com ack sobreposto. */
                nbuffed = nbuffed - 1;
                /* um quadro a menos no buffer */
                parar o cronômetro (ack esperado);
                /* O quadro chegou intacto; parar o cronômetro */
                inc (ack esperado);
                /* janela do remetente do contrato */
            }
            quebrar;
            case cksum err: break;
            /* apenas ignore os frames ruins */
            tempo limite do caso:
                /* problema; retransmitir todos os quadros pendentes */
                próximo quadro a ser enviado = ack esperado;
                /* comece a retransmitir aqui */
                para (i = 1; i <= nbuffed; i++) {
                    enviar dados (próximo quadro a enviar, quadro esperado, buffer); /* reenviar quadro */
                    inc (próximo quadro a ser enviado);
                    /* prepare-se para enviar o próximo */
                }
            }
            if (nbuffed < MAX SEQ)
                habilitar camada de rede ();
            outro
            desabilitar camada de rede ();
        }
    }
}

```

Figura 3-19. Um protocolo de janela deslizante usando go-back-n.

A questão é esta: todos os oito quadros pertencentes ao segundo lote chegaram com sucesso sem sucesso, ou todos os oito se perderam (contando os descartes após um erro como perdidos)? Em ambos os casos, o receptor enviaria o quadro 7 como confirmação.

238

A CAMADA DE LINK DE DADOS INDIVÍDUO. 3

O remetente não tem como saber. Por este motivo, o número máximo de os quadros de pé devem ser restritos a *MAX SEQ*.

Embora o protocolo 5 não armazene em buffer os quadros que chegam após um erro, ele não escapar completamente do problema de armazenamento em buffer. Já que um remetente pode ter que

retransmitir todos os frames não reconhecidos em um momento futuro, ele deve se agarrar a todos quadros transmitidos até que tenha certeza de que foram aceitos pelo receiver. Quando uma confirmação chega para o quadro n , quadros $n - 1, n - 2$, e assim por diante também são reconhecidos automaticamente. Este tipo de reconhecimento é chamado de **confirmação cumulativa**. Esta propriedade é especialmente importante quando alguns dos quadros anteriores de confirmação foram perdidos ou danificados sangrou. Sempre que chega qualquer confirmação, a camada de enlace de dados verifica para ver se algum buffer pode agora ser liberado. Se os buffers podem ser liberados (ou seja, há algum quarto disponível na janela), uma camada de rede previamente bloqueada agora pode ser baixado para causar mais eventos *prontos para a camada de rede*.

Para este protocolo, presumimos que sempre há tráfego reverso no qual confirmações nas costas. O Protocolo 4 não precisa dessa suposição, pois envia de volta um quadro toda vez que recebe um quadro, mesmo que já tenha enviado esse quadro. No próximo protocolo, resolveremos o problema de tráfego unilateral em um maneira elegante.

Como o protocolo 5 tem vários quadros pendentes, ele logicamente precisa de vários temporizadores, um por quadro excepcional. Cada frame expira independentemente de todos Os outros. No entanto, todos esses temporizadores podem ser facilmente simulados no software usando um único relógio de hardware que causa interrupções periódicas. O pendente timeouts formam uma lista vinculada, com cada nó da lista contendo o número de o relógio avança até que o cronômetro expire, o quadro sendo cronometrado e um ponteiro para o próximo nó.

```

10: 00: 00.000
10: 00: 00.005
5
1
8
2
6
3
6
3
8
2
Real
Tempo
Ponteiro para o próximo tempo limite
Quadro sendo cronometrado
Carrapatos para ir
(uma)
(b)

```

Figura 3-20. Simulação de vários temporizadores em software. (a) O tempo na fila saídas. (b) A situação após o término do primeiro tempo limite.

Como ilustração de como os temporizadores podem ser implementados, considere o exemplo da Fig. 3-20 (a). Suponha que o relógio avance uma vez a cada 1 ms. Inicialmente,

SEC. 3,4 PROTOCOLOS DE JANELA DE DESLIZAMENTO

239

o tempo real é 10: 00: 00.000; três tempos limite estão pendentes, às 10: 00: 00.005, 10: 00: 00.013 e 10: 00: 00.019. Cada vez que o relógio do hardware bate, o verdadeiro a hora é atualizada e o contador de tiques no topo da lista é diminuído. Quando

o contador de tiques torna-se zero, um tempo limite é causado e o nó é removido da lista, conforme mostrado na Fig. 3-20 (b). Embora esta organização exija que a lista ser verificado quando o *cronômetro de início* ou de *parada* é chamado, não exige muito trabalho por carapato. No protocolo 5, ambas as rotinas receberam um parâmetro indicando qual quadro deve ser cronometrado.

3.4.3 Um protocolo usando repetição seletiva

O protocolo go-back-n funciona bem se os erros forem raros, mas se a linha for ruim, desperdiça muita largura de banda em quadros retransmitidos. Uma estratégia alternativa, o protocolo de repetição seletiva, é permitir que o receptor aceite e armazene os quadros seguindo um danificado ou perdido.

Neste protocolo, tanto o remetente quanto o receptor mantêm uma janela de números de sequência aceitáveis, respectivamente. O tamanho da janela do remetente começa saí em 0 e cresce até algum máximo predefinido. A janela do receptor, em contraste, é sempre de tamanho fixo e igual ao máximo pré-determinado. Lá-

cever tem um buffer reservado para cada número de sequência dentro de sua janela fixa.

Associado a cada buffer está um bit (*chegado*) informando se o buffer está cheio ou vazio. Sempre que um quadro chega, seu número de sequência é verificado pela função *entre* para ver se ele cai dentro da janela. Se sim e se ainda não tiver sido recebido, é aceito e armazenado. Esta ação é realizada independentemente de o quadro não contém o próximo pacote esperado pela camada de rede. Claro, deve ser mantido dentro da camada de enlace de dados e não passado para a camada de rede até todos os quadros de numeração inferior já foram entregues à camada de rede na ordem correta. Um protocolo usando esse algoritmo é apresentado na Figura 3-21.

A recepção não sequencial introduz outras restrições no número de sequência de quadros comparados a protocolos nos quais os quadros são aceitos apenas em ordem. Podemos ilustrar o problema mais facilmente com um exemplo. Suponha que temos um 3-bit número de sequência, para que o remetente tenha permissão para transmitir até sete quadros antes de ser obrigado a aguardar uma confirmação. Inicialmente, o remetente e as janelas do receptor são mostradas na Fig.3-22 (a). O remetente agora transmite quadros de 0 a 6. A janela do receptor permite aceitar qualquer quadro com um número de sequência entre 0 e 6 inclusive. Todos os sete frames chegam corretamente, então o receptor os reconhece e avança sua janela para permitir o recebimento de 7, 0, 1, 2, 3, 4 ou 5, como mostrado na Fig. 3-22 (b). Todos os sete buffers são marcados como vazios. É neste ponto que o desastre atinge a forma de um raio atingindo o poste de telefone e apagando todos os reconhecimentos. O protocolo deve operar corretamente, apesar deste desastre. O remetente eventualmente atinge o tempo limite e volta a

transmite o quadro 0. Quando este quadro chega ao receptor, é feita uma verificação para ver se cair dentro da janela do receptor. Infelizmente, na Fig. 3-22 (b) o quadro 0 é

```
/* Protocolo 6 (repetição seletiva) aceita frames fora de ordem, mas passa pacotes para o
camada de rede em ordem. Associado a cada quadro pendente está um cronômetro. Quando o cronômetro
expira, apenas aquele quadro é retransmitido, não todos os quadros pendentes, como no protocolo 5. */
#define MAX SEQ 7
/* deve ser 2^n - 1 */
#define NR BUFS ((MAX SEQ + 1) / 2)
typedef enum {chegada de quadro, cksum err, tempo limite, camada de rede pronta, tempo limite de confirmação} tipo de
evento;
#include "protocol.h"
boolean no nak = true;
/* nenhum nak foi enviado ainda */
seq nr quadro mais antigo = MAX SEQ + 1;
/* o valor inicial é apenas para o simulador */
booleano estático entre (seq nr a, seq nr b, seq nr c)
```

```

{
/* O mesmo que no protocolo 5, mas mais curto e mais obscuro. */
return ((a <= b) && (b <c)) || ((c <a) && (a <= b)) || ((b <c) && (c <a));
}
estático void send frame (frame kind fk, seq nr frame nr, seq nr frame esperado, pacote buffer [])
{
/* Construir e enviar um quadro de dados, ack ou nak. */
frame s;
/* variável de rascunho */
s.kind = fk;
/* kind == data, ack ou nak */
if (fk == data) s.info = buffer [frame nr% NR BUFS];
s.seq = frame nr;
/* apenas significativo para frames de dados */
s.ack = (quadro esperado + MAX SEQ)% (MAX SEQ + 1);
if (fk == nak) não nak = false;
/* um nak por quadro, por favor */
para a (s) camada (s) física (s);
/* transmitir o quadro */
if (fk == data) iniciar o temporizador (frame nr% NR BUFS);
parar cronômetro de confirmação ();
/* sem necessidade de quadro de reconhecimento separado */
}
void protocol6 (void)
{
seq nr ack esperado;
/* borda inferior da janela do remetente */
seq nr próximo quadro a ser enviado;
/* borda superior da janela do remetente + 1 */
seq nr quadro esperado;
/* borda inferior da janela do receptor */
seq nr muito longe;
/* borda superior da janela do receptor + 1 */
int i;
/* índice no buffer pool */
frame r;
/* variável de rascunho */
pacote de saída buf [NR BUFS];
/* buffers para o fluxo de saída */
pacote em buf [NR BUFS];
/* buffers para o fluxo de entrada */
chegou booleano [NR BUFS];
/* mapa de bits de entrada */
seq nr nbuffed;
/* quantos buffers de saída usados atualmente */
evento de tipo de evento;
habilitar camada de rede ();
/* inicializar */
ack esperado = 0;
/* próxima confirmação esperada no fluxo de entrada */
próximo quadro a ser enviado = 0;
/* número do próximo quadro de saída */
quadro esperado = 0;
longe demais = NR BUFS;
nbuffed = 0;
/* inicialmente nenhum pacote é armazenado em buffer */
para (i = 0; i <NR BUFS; i++) chegou [i] = falso;
}

```

241

```
enquanto (verdadeiro) {
esperar pelo evento (& evento);
/* cinco possibilidades: ver tipo de evento acima */
switch (evento) {
camada de rede do caso pronta:
/* aceitar, salvar e transmitir um novo quadro */
nbuffered = nbuffered + 1;
/* expandir a janela */
da camada de rede (& out buf [próximo quadro para enviar% NR BUFS]); /* buscar novo pacote */
enviar quadro (dados, próximo quadro a enviar, quadro esperado, saída buf); /* transmitir o quadro */
inc (próximo quadro a ser enviado);
/* avançar borda superior da janela */
quebrar;
chegada da moldura da caixa:
/* um dado ou quadro de controle chegou */
da camada física (& r);
/* buscar quadro de entrada da camada física */
if (r.kind == data) {
/* Um quadro não danificado chegou. */
if ((r.seq! = frame esperado) && no nak)
enviar quadro (nak, 0, quadro esperado, out buf); senão inicie o cronômetro de confirmação ();
if (entre (quadro esperado, r.seq, muito longe) && (chegou [r.seq% NR BUFS] == falso)) {
/* Os quadros podem ser aceitos em qualquer ordem. */
chegou [r.seq% NR BUFS] = verdadeiro;
/* marcar buffer como cheio */
em buf [r.seq% NR BUFS] = r.info;
/* inserir dados no buffer */
enquanto (chegou [quadro esperado% NR BUFS]) {
/* Passar quadros e janela de avanço. */
para a camada de rede (& em buf [quadro esperado% NR BUFS]);
não nak = verdadeiro;
chegou [quadro esperado% NR BUFS] = falso;
inc (quadro esperado); /* avançar borda inferior da janela do receptor */
inc (muito longe);
/* avançar borda superior da janela do receptor */
iniciar cronômetro de confirmação ();
/* para ver se um ack separado é necessário */
}
}
}
if ((r.kind == nak) && entre (ack esperado, (r.ack + 1)% (MAX SEQ + 1), próximo quadro a enviar))
enviar quadro (dados, (r.ack + 1)% (MAX SEQ + 1), quadro esperado, saída buf);
while (entre (ack esperado, r.ack, próximo quadro a ser enviado)) {
nbuffered = nbuffered - 1;
/* lidar com ack sobreposto */
parar o cronômetro (ack esperado% NR BUFS);
/* quadro chegou intacto */
inc (ack esperado);
/* avançar borda inferior da janela do remetente */
}
quebrar;
case cksum err:
if (no nak) envia quadro (nak, 0, quadro esperado, out buf); /* quadro danificado */
quebrar;
tempo limite do caso:
enviar quadro (dados, quadro mais antigo, quadro esperado, buf de saída); /* atingimos o tempo limite */
quebrar;
tempo limite de confirmação de caso:
enviar quadro (ack, 0, quadro esperado, saída buf);
/* cronômetro de confirmação expirado; enviar confirmação */
}
if (nbuffered < NR BUFS) habilita a camada de rede (); caso contrário, desativa a camada de rede ();
```

```
}
```

Figura 3-21. Um protocolo de janela deslizante usando repetição seletiva.

242

A CAMADA DE LINK DE DADOS

INDIVÍDUO. 3

dentro da nova janela, por isso é aceito como um novo quadro. O receptor também envia um reconhecimento (associado) para o quadro 6, uma vez que 0 a 6 foram received.

O remetente fica feliz em saber que todos os quadros transmitidos realmente chegaram corretamente, de modo que avança sua janela e envia imediatamente os quadros 7, 0, 1, 2, 3, 4, e 5. Frame 7 será aceito pelo receptor e seu pacote será passado diretamente para a camada de rede. Imediatamente depois disso, a camada de enlace de dados de recebimento

verifica se já tem um frame 0 válido, descobre que tem e passa o antigo pacote em buffer para a camada de rede como se fosse um novo pacote. Consequentemente, a camada de rede obtém um pacote incorreto e o protocolo falha.

A essência do problema é que, após o receptor avançar sua janela, o novo intervalo de números de sequência válidos se sobreponha ao antigo. Consequentemente, o lote seguinte de frames pode ser duplicado (se todas as confirmações foram perdidas) ou novos (se todos os agradecimentos foram recebidos). Os pobres receiver não tem como distinguir esses dois casos.

A saída para este dilema consiste em certificar-se de que, após o receptor avançou sua janela, não há sobreposição com a janela original. Para garantir que não há sobreposição, o tamanho máximo da janela deve ser no máximo metade do intervalo de os números de sequência. Essa situação é mostrada na Fig. 3-22 (c) e na Fig. 3-22 (d).

Com 3 bits, os números de sequência variam de 0 a 7. Apenas quatro não reconhecidos os quadros devem estar pendentes a qualquer momento. Dessa forma, se o receptor acabou de aceitou os quadros de 0 a 3 e avançou sua janela para permitir a aceitação de quadros 4 a 7, ele pode dizer sem ambigüidade se os quadros subsequentes são retransmitidos sôes (0 a 3) ou novas (4 a 7). Em geral, o tamanho da janela para o protocolo 6 será ($MAX SEQ + 1$) / 2.

Uma questão interessante é: quantos buffers o receptor deve ter? Sob em nenhuma condição ele aceitará quadros cujos números de sequência estejam abaixo do borda inferior da janela ou frames cujos números de sequência estão acima do borda da janela. Consequentemente, o número de buffers necessários é igual ao tamanho da janela, não para o intervalo de números de sequência. No exemplo anterior de um número de seqüência de 3 bits, quatro buffers, numerados de 0 a 3, são necessários. Quando o quadro i chega, ele é colocado no buffer $i \bmod 4$. Observe que embora i e $(i + 4) \bmod 4$ estejam "competindo" pelo mesmo buffer, eles nunca estão dentro da janela no mesmo tempo, porque isso implicaria em um tamanho de janela de pelo menos 5.

Pelo mesmo motivo, o número de temporizadores necessários é igual ao número de buffers, não para o tamanho do espaço de sequência. Efetivamente, um cronômetro está associado com cada buffer. Quando o tempo se esgota, o conteúdo do buffer é retransmitido mitted.

O protocolo 6 também relaxa a suposição implícita de que o canal é fortemente carregado. Fizemos essa suposição no protocolo 5, quando contamos com os quadros sendo enviado na direção inversa para adicionar as confirmações. Se láverso o tráfego é leve, os reconhecimentos podem ser retidos por um longo período de tempo, o que pode causar problemas. No extremo, se houver muito tráfego em um

Remetente
 Receptor
 0 1 2 3 4 5 6 7
 0 1 2 3 4 5 6 7
 0 1 2 3 4 5 6 7
 0 1 2 3 4 5 6 7
 0 1 2 3 4 5 6 7
 0 1 2 3 4 5 6 7
 0 1 2 3 4 5 6 7
 0 1 2 3 4 5 6 7
 (uma)
 (b)
 (c)
 (d)

Figura 3-22. (a) Situação inicial com uma janela de tamanho 7. (b) Após 7 frames foram enviados e recebidos, mas não confirmados. (c) Situação inicial com um tamanho da janela de 4. (d) Após 4 frames terem sido enviados e recebidos, mas não reconhecido.

direção e nenhum tráfego na outra direção, o protocolo será bloqueado quando o a janela do remetente atinge seu máximo.

Para relaxar essa suposição, um cronômetro auxiliar é iniciado por *iniciar cronômetro de confirmação* após

chega um quadro de dados em sequência. Se nenhum tráfego reverso se apresentou antes este temporizador expira, um quadro de confirmação separado é enviado. Uma interrupção devida a o temporizador auxiliar é chamado de evento de *tempo limite de confirmação*. Com este arranjo, o tráfego

fluxo em apenas uma direção é possível porque a falta de quadros de dados reversos em quais reconhecimentos podem ser adicionados não é mais um obstáculo. Apenas um existe temporizador auxiliar, e se o *temporizador de confirmação inicial* for chamado enquanto o temporizador estiver em execução,

não tem efeito. O cronômetro não é zerado ou estendido, pois sua finalidade é fornecer alguma taxa mínima de reconhecimentos.

É essencial que o tempo limite associado ao temporizador auxiliar seja adequado habilmente mais curto do que o tempo limite usado para o tempo limite de quadros de dados. Esta condição é

necessário para garantir que um quadro recebido corretamente seja reconhecido com antecedência suficiente

que o temporizador de retransmissão do quadro não expire e retransmita o quadro.

O protocolo 6 usa uma estratégia mais eficiente do que o protocolo 5 para lidar com rors. Sempre que o receptor tem razão para suspeitar que ocorreu um erro, ele envia um quadro de confirmação negativa (NAK) de volta ao remetente. Tal frame é um pedido de retransmissão do frame especificado no NAK . Em dois casos, o receptor deve suspeitar: quando chega um quadro danificado ou um quadro diferente do esperado (quadro perdido potencial). Para evitar fazer vários todos os pedidos de retransmissão do mesmo quadro perdido, o receptor deve manter rastrear se um NAK já foi enviado para um determinado quadro. A variável *nenhum nak* no protocolo 6 é *verdadeiro* se nenhum NAK foi enviado ainda para o *quadro esperado* . E se

o NAK é mutilado ou perdido, nenhum dano real é causado, uma vez que o remetente irá eventualmente

ally tempo limite e retransmitir o quadro ausente de qualquer maneira. Se o quadro errado estiver chega depois que um NAK foi enviado e perdido, *nenhum nak* será *verdadeiro* e o auxiliar cronômetro será iniciado. Quando expirar, um ACK será enviado para resincronizar o remetente ao status atual do receptor.

ligeiramente maior do que o intervalo de tempo normal esperado entre o envio de um quadro e recebendo seu reconhecimento. NAKs não são úteis neste caso.

No entanto, em outras situações, o tempo pode ser altamente variável. Por exemplo, se o tráfego reverso é esporádico, o tempo antes da confirmação será menor quando há tráfego reverso e mais tempo quando não há. O remetente está de frente com a escolha de definir o intervalo para um valor pequeno (e arriscar desnecessariamente retransmissões sárias) ou configurá-lo para um valor alto (e ficar ocioso por um longo tempo após um erro). Ambas as opções desperdiçam largura de banda. Em geral, se o padrão o desvio do intervalo de confirmação é grande em comparação com o próprio intervalo, o cronômetro é definido como "solto" para ser conservador. NAKs podem, então, acelerar consideravelmente a retransmissão de quadros perdidos ou danificados.

Intimamente relacionado à questão dos tempos limite e NAKs está a questão de determinar minerando qual quadro causou um tempo limite. No protocolo 5, é sempre considerado como *esperado*,

porque é sempre o mais antigo. No protocolo 6, não há uma maneira trivial de determinar quem expirou. Suponha que os quadros de 0 a 4 tenham sido transmitidos, o que significa que a lista de quadros pendentes é 01234, na ordem do mais antigo para o mais novo.

Agora imagine que 0 tempo limite, 5 (um novo quadro) é transmitido, 1 tempo limite, 2 vezes para fora, e 6 (outro novo quadro) é transmitido. Neste ponto, a lista de pendentes frames é 3405126, do mais antigo ao mais novo. Se todo o tráfego de entrada (ou seja, reconhecer quadros de suporte de borda) se perder por um tempo, os sete quadros pendentes tempo limite nessa ordem.

Para evitar que o exemplo fique ainda mais complicado do que já é, não mostramos a administração do cronômetro. Em vez disso, apenas assumimos que o *quadro mais antigo* variável é definido no tempo limite para indicar qual quadro atingiu o tempo limite.

3.5 EXEMPLO DE PROTOCOLOS DE LINK DE DADOS

Dentro de um único edifício, as LANs são amplamente utilizadas para interconexão, mas a maioria a infraestrutura de rede de área ampla é construída a partir de linhas ponto a ponto. No cap. 4, veremos as LANs. Aqui, examinaremos os protocolos de link de dados encontrados em linhas ponto a ponto na Internet em duas situações comuns. A primeira situação é quando os pacotes são enviados por links de fibra óptica SONET em redes de longa distância. Esses links são amplamente usados, por exemplo, para conectar roteadores em diferentes locais da rede de um ISP.

A segunda situação é para links ADSL rodando no loop local do telefone telefônica na borda da Internet. Esses links conectam milhões de indivíduos uais e empresas para a Internet.

A Internet precisa de links ponto a ponto para esses usos, bem como de modo dial-up dems, linhas alugadas e modems a cabo e assim por diante. Um protocolo padrão chamado PPP

SEC. 3,5

EXEMPLO DE PROTOCOLOS DE LINK DE DADOS

245

(**Protocolo ponto a ponto**) é usado para enviar pacotes por meio desses links. PPP é de-mulgado no RFC 1661 e posteriormente elaborado no RFC 1662 e outros RFCs (Simpson, 1994a, 1994b). Os links SONET e ADSL aplicam-se ao PPP, mas de maneiras diferentes.

3.5.1 Pacote sobre SONET

SONET, que abordamos na Seç. 2.6.4, é o protocolo da camada física que é mais comumente usado sobre os links de fibra óptica de área ampla que compõem o backbone das redes de comunicação, incluindo o sistema telefônico. Fornece um bitstream que funciona a uma taxa bem definida, por exemplo 2,4 Gbps para um link OC-48. Este fluxo de bits é organizado como payloads de bytes de tamanho fixo que se repetem a cada 125 µs, se há ou não dados do usuário para enviar.

Para transportar pacotes através desses links, algum mecanismo de enquadramento é necessário para

distinguir pacotes ocasionais do fluxo de bits contínuo em que estão transportado. O PPP é executado em roteadores IP para fornecer esse mecanismo, conforme mostrado em

Fig. 3-23.



Figura 3-23. Pacote sobre SONET. (a) Uma pilha de protocolo. (b) Relacionamentos de estrutura.

O PPP é aprimorado em relação a um protocolo anterior e mais simples chamado **SLIP (Serial Line Inter-**

protocolo de rede) e é usado para lidar com a configuração do link de detecção de erros, suporte vários protocolos, permitem autenticação e muito mais. Com um amplo conjunto de opções, O PPP oferece três recursos principais:

1. Um método de enquadramento que delinea inequivocamente o final de um quadro e o início do próximo. O formato do quadro também lida com detecção de erro.
2. Um protocolo de controle de link para trazer linhas, testá-las, negociar opções de seleção e diminuí-las novamente quando não é mais necessário. Este protocolo é denominado **LCP (Link Control Protocol).**
3. Uma maneira de negociar as opções da camada de rede de uma forma que seja independente do protocolo da camada de rede a ser usado. O método escolhido é ter um **NCP (protocolo de controle de rede)** diferente para cada rede camada de trabalho suportada.

Página 270

246

A CAMADA DE LINK DE DADOS

INDIVÍDUO. 3

O formato de quadro PPP foi escolhido para se parecer muito com o formato de quadro de **HDLC (High-level Data Link Control)**, uma instância amplamente usada de um anterior família de protocolos, já que não havia necessidade de reinventar a roda.

A principal diferença entre PPP e HDLC é que o PPP é orientado por bytes ao invés de bit orientado. Em particular, PPP usa byte stuffing e todos os frames são um número integral de bytes. HDLC usa bit stuffing e permite quadros de, digamos, 30,25 bytes.

Há uma segunda grande diferença na prática, entretanto. HDLC fornece re-transmissão responsável com uma janela deslizante, confirmações e tempos limite no maneira que estudamos. O PPP também pode fornecer transmissão confiável em ambientes ruidosos ambientes, como redes sem fio; os detalhes exatos são definidos no RFC 1663.

No entanto, isso raramente é feito na prática. Em vez disso, um " modo não numerado " é quase Sempre usado na Internet para fornecer serviço não reconhecido sem conexão.

O formato do quadro PPP é mostrado na Figura 3-24. Todos os quadros PPP começam com o byte de sinalizador HDLC padrão de 0x7E (0111110). O byte de sinalização é preenchido se ocorrer

dentro do campo *Payload* usando o byte de escape 0x7D. O seguinte byte é o byte de escape XORed com 0x20, que inverte o 5º bit. Por exemplo, 0x7D 0x5E é a sequência de escape para o byte 0x7E do sinalizador. Isso significa o início e o fim de quadros podem ser pesquisados simplesmente procurando o byte 0x7E, já que não ocorrer em outro lugar. A regra de destuffing ao receber um quadro é procurar 0x7D,

remova-o e execute um XOR no byte seguinte com 0x20. Além disso, apenas um byte de sinalizador é necessário entre os quadros. Vários bytes de sinalização podem ser usados para preencher o link quando houver não há frames a serem enviados.

Após o byte do sinalizador de início do quadro, vem o campo *Endereço*. Este campo é tudo formas definidas para o valor binário 11111111 para indicar que todas as estações devem aceitar o quadro, Armação. Usar esse valor evita o problema de atribuir endereços de link de dados.

Bandeira	01111110
Bandeira	01111110
Endereço	11111111
Protocolo	Ao controle
	00000011
Carga útil	
Checksum	
Bytes	1
	1
	1 ou 2
	1
Variável	2 ou 4
	1

Figura 3-24. O formato de quadro completo PPP para operação em modo não numerado.

O campo *Endereço* é seguido pelo campo *Controle*, o valor padrão do qual é 00000011. Este valor indica um quadro não numerado.

Uma vez que os campos *Endereço* e *Controle* são sempre constantes na configuração padrão configuração, LCP fornece o mecanismo necessário para as duas partes negociar uma opção para omiti-los completamente e economizar 2 bytes por quadro.

O quarto campo PPP é o campo *Protocolo*. Seu trabalho é dizer que tipo de pacote está no campo *Payload*. Os códigos que começam com um bit 0 são definidos para IP versão 4, IP versão 6 e outros protocolos de camada de rede que podem ser usados, como IPX e

Página 271

SEC. 3,5

EXEMPLO DE PROTOCOLOS DE LINK DE DADOS

247

AppleTalk. Os códigos que começam com 1 bit são usados para protocolos de configuração PPP, incluindo LCP e um NCP diferente para cada protocolo de camada de rede suportado.

O tamanho padrão do campo *Protocolo* é de 2 bytes, mas pode ser negociado até 1 byte usando LCP. Os designers foram talvez excessivamente cautelosos ao pensar que algum dia, pode haver mais de 256 protocolos em uso.

O campo *Payload* tem comprimento variável, até o máximo negociado. Se o comprimento não é negociado usando LCP durante a configuração da linha, um comprimento padrão de 1500

bytes é usado. O preenchimento pode seguir a carga útil, se necessário.

Depois do campo *Payload*, vem o campo *Checksum*, que normalmente tem 2 bytes, mas uma soma de verificação de 4 bytes pode ser negociada. A soma de verificação de 4 bytes é na verdade a mesma

CRC de 32 bits cujo polinômio gerador é dado no final da Seção 3.2.2. O 2-A soma de verificação de bytes também é um CRC padrão da indústria.

PPP é um mecanismo de enquadramento que pode transportar os pacotes de vários protocolos em muitos tipos de camadas físicas. Para usar PPP em vez de SONET, as escolhas a fazer são descritos na RFC 2615 (Malis e Simpson, 1999). Uma soma de verificação de 4 bytes é usado, uma vez que este é o principal meio de detecção de erros de transmissão ao longo do camadas físicas, de link e de rede. Recomenda-se que o *endereço*, *controle*, e os campos de *protocolo* não podem ser compactados, uma vez que os links SONET já são executados em taxas muito altas.

Existe também uma característica incomum. A carga útil PPP é embaralhada (conforme descrito na seção 2.5.1) antes de ser inserido na carga SONET. Scrambling XORs o carga útil com uma longa sequência pseudo-aleatória antes de ser transmitida. O problema é que o fluxo de bits SONET precisa de transições de bits frequentes para a sincronização. Essas transições acontecem naturalmente com a variação nos sinais de voz, mas nos dados comunicação, o usuário escolhe as informações que são enviadas e pode enviar um pacote com uma longa duração de 0s. Com o embaralhamento, a probabilidade de um usuário ser capaz causar problemas enviando uma longa sequência de 0s torna-se extremamente baixa.

Antes que os quadros PPP possam ser transportados por linhas SONET, o link PPP deve ser estabelecido e configurado. As fases pelas quais o link passa quando é trazido para cima, usado e retirado novamente são mostrados na Fig. 3-25.

O link começa no estado *DEAD*, o que significa que não há conexão em a camada física. Quando uma conexão de camada física é estabelecida, o link move-se para *ESTABELECER*. Neste ponto, os pares PPP trocam uma série de LCP pacotes, cada um transportado no campo *Payload* de um quadro PPP, para selecionar a opção PPP para o link das possibilidades mencionadas acima. O par iniciador apresenta opções, e o par que responde as aceita ou rejeita, no todo ou parte. O respondente também pode fazer propostas alternativas.

Se a negociação da opção LCP for bem-sucedida, o link chega a *AUTENTICAR* Estado. Agora as duas partes podem verificar a identidade uma da outra, se desejado. E se a autenticação for bem-sucedida, o estado *NETWORK* é inserido e uma série de NCP os pacotes são enviados para configurar a camada de rede. É difícil generalizar sobre os protocolos NCP porque cada um é específico para algum protocolo da camada de rede e permite que sejam feitas solicitações de configuração específicas para esse protocolo.

Página 272

248

A CAMADA DE LINK DE DADOS

INDIVÍDUO. 3

REDE
MORTO
TERMINAR
ABRIR
ESTABELECER
AUTENTICAR
Transportadora
detectou
Ambos os lados
concordar com as opções
Autenticação
bem sucedido
NCP
configuração
Transportadora
desistiu
Falhou
Falhou
Feito

Figura 3-25. Diagrama de estado para ativar e desativar um link PPP.

Para IP, por exemplo, a atribuição de endereços IP a ambas as extremidades do link é o mais importante.

Uma vez que *OPEN* é alcançado, o transporte de dados pode ocorrer. É neste estado que o IP os pacotes são transportados em quadros PPP através da linha SONET. Quando o transporte de dados é terminado, o link passa para o estado *TERMINATE* e, a partir daí, volta para o estado *DEAD* quando a conexão da camada física é interrompida.

3.5.2 ADSL (Asymmetric Digital Subscriber Loop)

ADSL conecta milhões de assinantes domésticos à Internet a megabit / s taxas sobre o mesmo loop local de telefone que é usado para o antigo serviço de telefone comum vice. Na seção 2.5.3, descrevemos como um dispositivo chamado modem DSL é adicionado o lado da casa. Ele envia bits pelo loop local para um dispositivo chamado DSLAM (DSL Access Multiplexer), pronunciado "dee-slam" no local da companhia telefônica

escritório. Agora vamos explorar com mais detalhes como os pacotes são transportados por ADSL links.

O quadro geral para os protocolos e dispositivos usados com ADSL é mostrado em Fig. 3-26. Protocolos diferentes são implantados em redes diferentes, então temos que escolher sen para mostrar o cenário mais popular. Dentro de casa, um computador como um PC envia pacotes IP para o modem DSL usando uma camada de link como Ethernet. DSL modem então envia os pacotes IP através do loop local para o DSLAM usando o protocolos que estamos prestes a estudar. No DSLAM (ou em um roteador conectado a ele dependendo da implementação) os pacotes IP são extraídos e entram em um ISP rede para que possam chegar a qualquer destino na Internet.

Os protocolos mostrados no link ADSL na Fig. 3-26 começam na parte inferior com a camada física ADSL. Eles são baseados em um esquema de modulação digital chamado

Página 273

SEC. 3,5

EXEMPLO DE PROTOCOLOS DE LINK DE DADOS

249

AAL5
ADSL
Local
ciclo
ATM
PPP
DSLAM
(com roteador)
AAL5
ADSL
ATM
PPP
DSL
modem
PC
Ethernet
Internet
Casa do cliente
Escritório do ISP
Ethernet
IP
Ligaçāo
IP

Figura 3-26. Pilhas de protocolo ADSL.

multiplexação por divisão de frequência ortogonal (também conhecido como multiton discreto), como

vimos na Seção 2.5.3. Perto do topo da pilha, logo abaixo da camada de rede IP, está PPP. Este protocolo é o mesmo PPP que acabamos de estudar para packet over Transportes SONET. Funciona da mesma forma para estabelecer e configurar o link e transportar pacotes IP.

Entre ADSL e PPP estão ATM e AAL5. Estes são novos protocolos que não vimos antes. **ATM (Modo de transferência assíncrona)** foi projetado no início de 1990 e lançado com incrível hype. Prometia uma tecnologia de rede tecnologia que resolveria os problemas mundiais de telecomunicações pela fusão voz, dados, televisão a cabo, telegrafo, pombo-correio, latas conectadas por cordas, tom toms e tudo mais em um sistema integrado que poderia fazer tudo para todos. Isso não aconteceu. Em grande parte, os problemas do ATM eram semelhantes aos que descrevemos em relação aos protocolos OSI, ou seja, tempo ruim , tecnologia, implementação e política. No entanto, ATM era muito mais bem-sucedido do que o OSI. Embora não tenha dominado o mundo, permanece amplo Usado em nichos, incluindo linhas de acesso de banda larga, como DSL e links WAN dentro das redes telefônicas.

ATM é uma camada de enlace que se baseia na transmissão de **células** de comprimento fixo de em formação. O " Assíncrono " em seu nome significa que as células nem sempre precisam ser enviados da maneira que os bits são enviados continuamente por linhas síncronas, como em SONET. As células só precisam ser enviadas quando houver informações para transportar. ATM é uma tecnologia orientada para conexão. Cada célula carrega um **circuito virtual**

identificador em seu cabeçalho e os dispositivos usam esse identificador para encaminhar células ao longo dos caminhos de conexões estabelecidas.

Cada uma das células tem 53 bytes de comprimento, consistindo em uma carga útil de 48 bytes mais uma carga de 5 bytes no cabeçalho. Ao usar células pequenas, o ATM pode dividir com flexibilidade a largura de banda de um corpo

link de camada de cal entre diferentes usuários em fatias finas. Esta habilidade é útil quando, por exemplo, enviar voz e dados por meio de um link sem ter dados longos pacotes que causariam grandes variações no atraso das amostras de voz. A escolha incomum para o comprimento da célula (por exemplo, em comparação com a escolha mais natural de um

Página 274

250

A CAMADA DE LINK DE DADOS
INDIVÍDUO. 3

poder de 2) é uma indicação de quão político era o design do ATM. O

Tamanho de 48 bytes para a carga útil foi um meio-termo para resolver um impasse entre A Europa, que queria células de 32 bytes, e os EUA, que queriam células de 64 bytes. Uma breve visão geral do ATM é fornecida por Siu e Jain (1995).

Para enviar dados por uma rede ATM, eles precisam ser mapeados em uma sequência de células. Este mapeamento é feito com uma camada de adaptação ATM em um processo chamado seg-

mentação e remontagem. Várias camadas de adaptação foram definidas para diferentes serviços diferentes, variando de amostras periódicas de voz a dados em pacote. O principal usado para dados de pacote é **AAL5** (ATM Adaptation Layer 5).

Um quadro AAL5 é mostrado na Figura 3-27. Em vez de um cabeçalho, tem um trailer que fornece o comprimento e tem um CRC de 4 bytes para detecção de erros. Naturalmente, o CRC é o mesmo usado para PPP e IEEE 802 LANs como Ethernet. Wang e Crowcroft (1992) mostrou que é forte o suficiente para detectar erros não tradicionais como reordenamento de células. Além de uma carga útil, o quadro AAL5 possui preenchimento. Isso arredonda o comprimento total para um múltiplo de 48 bytes para que o quadro pode ser dividido uniformemente em células. Nenhum endereço é necessário no quadro, pois o identificador de circuito real transportado em cada célula o levará ao destino correto.

Protocolo PPP
Carga útil PPP
Almofada
Não utilizado
comprimento
CRC
Bytes
1 ou 2
0 a 47
2
2
4
Trailer AAL5
Variável
Carga útil AAL5

Figura 3-27. Quadro AAL5 com dados PPP.

Agora que descrevemos o ATM, temos apenas que descrever como o PPP torna uso de ATM no caso de ADSL. Isso é feito com outro padrão chamado **PPPoA** (PPP sobre ATM). Este padrão não é realmente um protocolo (por isso não aparecem na Fig. 3-26), mas mais uma especificação de como trabalhar com PPP e Quadros AAL5. É descrito em RFC 2364 (Gross et al., 1998).

Apenas o protocolo PPP e os campos de carga útil são colocados na carga útil AAL5, como mostrado na Fig. 3-27. O campo do protocolo indica ao DSLAM na extremidade oposta se a carga útil é um pacote IP ou um pacote de outro protocolo, como LCP. A extremidade oposta sabe que as células contêm informações PPP porque um ATM circuito virtual é configurado para este propósito.

Dentro do quadro AAL5, o enquadramento PPP não é necessário, pois não serviria para

pose; ATM e AAL5 já fornecem o enquadramento. Mais enquadramento seria inútil. O PPP CRC também não é necessário porque AAL5 já inclui o mesmo CRC. Este mecanismo de detecção de erros complementa o ADSL físico codificação de camada de um código Reed-Solomon para correção de erros e um CRC de 1 byte para a detecção de quaisquer erros restantes não detectados de outra forma. Este esquema tem um mecanismo de recuperação de erro muito mais sofisticado do que quando os pacotes são enviados em uma linha SONET porque ADSL é um canal muito mais ruidoso.

Página 275

SEC. 3,6
RESUMO

251

3.6 RESUMO

A tarefa da camada de enlace de dados é converter o fluxo de bits bruto oferecido pelo camada física em um fluxo de quadros para uso pela camada de rede. A camada de link pode apresentar este fluxo com vários níveis de confiabilidade, variando de serviço sem conexão e não reconhecido para serviço confiável e orientado para conexão. Vários métodos de enquadramento são usados, incluindo contagem de bytes, preenchimento de bytes e bit estofamento. Os protocolos de enlace de dados podem fornecer controle de erros para detectar ou corrigir danos quadros抗igos e retransmitir quadros perdidos. Para evitar que um remetente rápido ultrapasse ning um receptor lento, o protocolo de enlace de dados também pode fornecer controle de fluxo. O sli- O mecanismo de janela ding é amplamente utilizado para integrar controle de erros e controle de fluxo de uma forma simples. Quando o tamanho da janela é 1 pacote, o protocolo é parar e esperar. Códigos para correção e detecção de erros adicionam informações redundantes às mensagens sábios usando uma variedade de técnicas matemáticas. Códigos convolucionais e Os códigos Reed-Solomon são amplamente implantados para correção de erros, com baixa densidade os códigos de verificação de paridade estão aumentando em popularidade. Os códigos para detecção de erros que são usados na prática incluem verificações e somas de verificação de redundância cílica. Todos esses códigos pode ser aplicado na camada de link, bem como na camada física e em camadas superiores. Examinamos uma série de protocolos que fornecem uma camada de link confiável usando conhecimentos e retransmissões, ou ARQ (Automatic Repeat reQuest), sob suposições mais realistas. Partindo de um ambiente livre de erros no qual o receptor pode lidar com qualquer quadro enviado a ele, introduzimos o controle de fluxo, seguido por controle de erros com números de sequência e o algoritmo de parar e esperar. Então nós usou o algoritmo de janela deslizante para permitir a comunicação bidirecional e introduzir o conceito de pegamento. Os dois últimos protocolos canalizam o transmissão de vários quadros para evitar que o remetente bloquee um link com um longo atraso de propagação. O receptor pode descartar todos os quadros, exceto o próximo em sequência, ou buffer de quadros fora de ordem e enviar confirmação negativa arestas para maior eficiência de largura de banda. A primeira estratégia é voltar e voltar protocolo, e a última estratégia é um protocolo de repetição seletiva. A Internet usa PPP como o principal protocolo de enlace de dados em linhas ponto a ponto. Ele fornece um serviço não confirmado sem conexão, usando bytes de sinalização para delimitar frames e um CRC para detecção de erros. É usado para transportar pacotes em uma gama de links, incluindo links SONET em redes de longa distância e links ADSL para o casa.

PROBLEMAS

1. Um pacote da camada superior é dividido em 10 quadros, cada um dos quais com 80% de chance de ser

riving sem danos. Se nenhum controle de erro for feito pelo protocolo de link de dados, quantos vezes a mensagem deve ser enviada, em média, para que tudo seja concluído?

Página 276

252

A CAMADA DE LINK DE DADOS

INDIVÍDUO. 3

2. A seguinte codificação de caracteres é usada em um protocolo de link de dados:

A: 01000111

B: 11100011

FLAG: 01111110

ESC: 11100000

Mostra a sequência de bits transmitida (em binário) para o quadro de quatro caracteres AB ESC

FLAG quando cada um dos seguintes métodos de enquadramento é usado:

(a) Contagem de bytes.

(b) Sinalizar bytes com preenchimento de bytes.

(c) Iniciando e terminando bytes de flag com preenchimento de bits.

3. O seguinte fragmento de dados ocorre no meio de um fluxo de dados para o qual o byte-

O algoritmo de enchimento descrito no texto é usado: AB ESC C ESC FLAG FLAG D.

Qual é a saída após o enchimento?

4. Qual é a sobrecarga máxima no algoritmo de preenchimento de bytes?

5. Um de seus colegas de classe, Scrooge, apontou que é um desperdício terminar cada quadro com um byte de sinalizador e, em seguida, comece o próximo com um segundo byte de sinalizador. Um byte de bandeira

poderia fazer o trabalho também, e um byte salvo é um byte ganho. Você concorda?

6. Uma sequência de bits, 0111101111101111110, precisa ser transmitida na camada de enlace de dados.

Qual é a string realmente transmitida após o enchimento de bits?

7. Você pode pensar em alguma circunstância em que um protocolo de malha aberta (por exemplo, um Hamming código) pode ser preferível aos protocolos do tipo feedback discutidos ao longo deste capítulo?

8. Para fornecer mais confiabilidade do que um único bit de paridade pode fornecer, uma codificação de detecção de erro

esquema usa um bit de paridade para verificar todos os bits ímpares e uma segunda paridade bit para todos os bits pares. Qual é a distância de Hamming deste código?

9. As mensagens de dezesseis bits são transmitidas usando um código de Hamming. Quantos bits de verificação são necessários para garantir que o receptor possa detectar e corrigir erros de bit único? Mostre o padrão de bits transmitido para a mensagem 1101001100110101. Suponha que parity é usado no código de Hamming.

10. Um código de Hamming de 12 bits cujo valor hexadecimal é 0xE4F chega a um receptor.

Qual era o valor original em hexadecimal? Suponha que não mais do que 1 bit está em erro.

11. Uma maneira de detectar erros é transmitir dados como um bloco de n linhas de k bits por linha e adicione bits de paridade a cada linha e coluna. O bit no canto inferior direito é um bit de paridade que verifica sua linha e sua coluna. Este esquema detectará todos os erros únicos? Erros duplos? Erros triplos? Mostre que este esquema não pode detectar algum erro de quatro bits.

12. Suponha que os dados sejam transmitidos em blocos de tamanhos 1000 bits. Qual é o máximo taxa de erro sob a qual o mecanismo de detecção e retransmissão de erros (1 bit de paridade por bloco) é melhor do que usar o código de Hamming? Suponha que os erros de bit sejam independentes de um ao outro e nenhum erro de bit ocorre durante a retransmissão.

13. Um bloco de bits com n linhas e colunas k usa bits de paridade horizontal e vertical para detecção de erro. Suponha que exatamente 4 bits sejam invertidos devido a erros de transmissão. Derive uma expressão para a probabilidade de que o erro não seja detectado.

Página 277

INDIVÍDUO. 3

PROBLEMAS

253

14. Usando o codificador convolucional da Fig. 3-7, qual é a sequência de saída quando a entrada a sequência é 10101010 (da esquerda para a direita) e o estado interno é inicialmente zero?

15. Suponha que uma mensagem 1001 1100 1010 0011 seja transmitida usando Internet Checksum (Palavra de 4 bits). Qual é o valor da soma de verificação?

16. Qual é o resto obtido dividindo $x^7 + x^5 + 1$ pelo polinômio gerador $x^3 + 1$?

- 17.** Um fluxo de bits 10011101 é transmitido usando o método CRC padrão descrito no texto. O polinômio gerador é $x^3 + 1$. Mostra a string de bits real transmitida. Suponha que o terceiro bit da esquerda seja invertido durante a transmissão. Mostre que isso é detectado no final do receptor. Dê um exemplo de erros de bits na sequência de bits transmitidos que não serão detectados pelo receptor.
- 18.** É enviada uma mensagem de 1024 bits que contém 992 bits de dados e 32 bits CRC. CRC é colocado usando o polinômio CRC de 32 graus padronizado IEEE 802. Para cada um dos a seguir, explique se os erros durante a transmissão da mensagem serão detectados por o receptor:
- Ocorreu um erro de bit único.
 - Havia dois erros de bit isolados.
 - Houve 18 erros de bits isolados.
 - Houve 47 erros de bit isolados.
 - Houve um erro de burst longo de 24 bits.
 - Houve um erro de burst longo de 35 bits.
- 19.** Na discussão do protocolo ARQ na Seção 3.3.3, um cenário foi delineado que resultou no receptor aceitar duas cópias do mesmo quadro devido a uma perda de reconhecimento moldura de borda. É possível que um destinatário aceite várias cópias do mesmo quadro quando nenhum dos quadros (mensagem ou confirmação) é perdido?
- 20.** Um canal tem uma taxa de bits de 4 kbps e um atraso de propagação de 20 ms. Para qual alcance de tamanhos de quadro, parar e esperar oferece uma eficiência de pelo menos 50%?
- 21.** No protocolo 3, é possível ao remetente iniciar o cronômetro quando ele já está em execução? Em caso afirmativo, como isso pode ocorrer? Se não, por que é impossível?
- 22.** Um tronco T1 de 3.000 km de comprimento é usado para transmitir quadros de 64 bytes usando o protocolo 5. Se a velocidade de propagação é de $6 \mu\text{s} / \text{km}$, quantos bits os números de sequência devem ter?
- 23.** Imagine um protocolo de janela deslizante usando tantos bits para números de sequência que o envoltório nunca ocorre. Quais relações devem ser mantidas entre as quatro bordas da janela e o tamanho da janela, que é constante e o mesmo para o remetente e o receptor?
- 24.** Se o procedimento *entre* no protocolo 5 verificado para a condição $a \leq b \leq c$ em vez de a condição $a \leq b < c$, teria algum efeito sobre a correção ou eficácia do protocolo ficção? Explique sua resposta.
- 25.** No protocolo 6, quando um quadro de dados chega, uma verificação é feita para ver se o número de sequência é diferente do esperado e *nenhum nak* é verdadeiro. Se ambas as condições forem mantidas, um NAK é enviado. Caso contrário, o temporizador auxiliar é iniciado. Suponha que a cláusula *else* fosse omitido. Esta mudança afetaria a correção do protocolo?

Página 278

254

A CAMADA DE LINK DE DADOS

INDIVÍDUO. 3

- 26.** Suponha que o loop while de três instruções perto do final do protocolo 6 foi removido do código. Isso afetaria a exatidão do protocolo ou apenas o desempenho? Explique sua resposta.
- 27.** A distância da Terra a um planeta distante é de aproximadamente $9 \times 10^{10} \text{ m}$. O que é utilização de canal se um protocolo parar e esperar for usado para transmissão de quadros em um 64 Link ponto a ponto de Mbps? Suponha que o tamanho do quadro seja de 32 KB e a velocidade da luz é $3 \times 10^8 \text{ m/s}$.
- 28.** No problema anterior, suponha que um protocolo de janela deslizante seja usado. Para quê tamanho da janela de envio a utilização do link será de 100%? Você pode ignorar o protocolo tempos de processamento no emissor e no receptor.
- 29.** No protocolo 6, o código para *chegada de quadro* tem uma seção usada para NAK s. Esta seção é chamado se o quadro recebido for um NAK e outra condição for atendida. Dê um cenário onde a presença dessa outra condição é essencial.
- 30.** Considere a operação do protocolo 6 em uma linha perfeita de 1 Mbps (isto é, livre de erros). o o tamanho máximo do quadro é de 1000 bits. Novos pacotes são gerados com 1 segundo de intervalo. o o intervalo de tempo limite é de 10 mseg. Se o temporizador de reconhecimento especial for eliminado, ocorreria tempo limite desnecessário. Quantas vezes a mensagem média seria transmitido?
- 31.** No protocolo 6, $\text{MAX SEQ} = 2^n - 1$. Embora esta condição seja obviamente desejável para fazer uso eficiente de bits de cabeçalho, não demonstramos que é essencial. Faz o protocolo funciona corretamente para $\text{MAX SEQ} = 4$, por exemplo?
- 32.** Quadros de 1000 bits são enviados por um canal de 1 Mbps usando um satélite geoestacionário cujo tempo de propagação da terra é 270 mseg. Agradecimentos são sempre pegado carona em quadros de dados. Os cabeçalhos são muito curtos. Sequência de três bits num-

bers são usados. Qual é a utilização máxima de canal alcançável para

- (a) Pare e espere?
- (b) Protocolo 5?
- (c) Protocolo 6?

33. Calcule a fração da largura de banda que é desperdiçada em overhead (cabeçalhos e retransmissões) para o protocolo 6 em um canal de satélite de 50 kbps altamente carregado com dados quadros consistindo em 40 cabeçalhos e 3960 bits de dados. Suponha que a propagação do sinal o tempo da Terra ao satélite é 270 mseg. Quadros ACK nunca ocorrem. Quadros NAK são 40 bits. A taxa de erro para frames de dados é de 1%, e a taxa de erro para frames NAK é insignificante. Os números de sequência são de 8 bits.

34. Considere um canal de satélite de 64 kbps sem erros usado para enviar quadros de dados de 512 bytes em uma direção, com reconhecimentos muito curtos voltando para o outro lado. O que é a taxa de transferência máxima para tamanhos de janela de 1, 7, 15 e 127? O satélite terrestre o tempo de propagação é 270 mseg.

35. Um cabo de 100 km de comprimento é executado na taxa de dados T1. A velocidade de propagação no cabo é 2/3 da velocidade da luz no vácuo. Quantos bits cabem no cabo?

36. Dê pelo menos uma razão pela qual o PPP usa o enchimento de bytes em vez de enchimento de bits para evitar bytes de sinalização acidentais dentro da carga útil de causar confusão.

Página 279

INDIVÍDUO. 3

PROBLEMAS

255

37. Qual é a sobrecarga mínima para enviar um pacote IP usando PPP? Conte apenas o head introduzido pelo próprio PPP, não a sobrecarga do cabeçalho IP. Qual é o máximo a sobrecarga?

38. Um pacote IP de 100 bytes é transmitido por um loop local usando a pilha de protocolo ADSL. Como muitas células ATM serão transmitidas? Descreva resumidamente seu conteúdo.

39. O objetivo deste exercício de laboratório é implementar um mecanismo de detecção de erros usando o algoritmo CRC padrão descrito no texto. Escreva dois programas, *gerador* e *verificador*. O programa *gerador* lê da entrada padrão uma linha de texto ASCII,

retendo uma mensagem de n bits consistindo em uma sequência de 0s e 1s. A segunda linha é k -bit polinomial, também em ASCII. Ele produz para a saída padrão uma linha de texto ASCII com $n + k$ 0s e 1s representando a mensagem a ser transmitida. Em seguida, ele produz o poli nominal, assim como lê. O programa *verificador* lê a saída do gerador

programa e emite uma mensagem indicando se está correto ou não. Finalmente, escreva um

programa, *alter*, que inverte 1 bit na primeira linha dependendo do seu argumento (o bit

número contando o bit mais à esquerda como 1), mas copia o resto das duas linhas corretamente.

Digitando

gerador <arquivo | verificador

você deve ver que a mensagem está correta, mas digitando

gerador <arquivo | alter arg | verificador

você deve receber a mensagem de erro.

Página 280

Esta página foi intencionalmente deixada em branco

Página 281

4

O CONTROLE DE ACESSO MÉDIO SUBLAYER

Os links de rede podem ser divididos em duas categorias: aqueles que usam ponto a ponto conexões e aqueles que usam canais de transmissão. Estudamos links ponto a ponto

no cap. 2; este capítulo trata de links de broadcast e seus protocolos.

Em qualquer rede de transmissão, a questão principal é como determinar quem pode usar o canal quando há competição por ele. Para fazer isso, considere uma chamada de conferência em que seis pessoas, em seis telefones diferentes, estão todas conectadas que cada um pode ouvir e falar com todos os outros. É muito provável que quando um dos eles param de falar, dois ou mais vão começar a falar ao mesmo tempo, levando ao caos. Em um encontro cara a cara, o caos é evitado por meios externos. Por exemplo, em um encontro, as pessoas levantam as mãos para pedir permissão para falar. Quando apenas um canal está disponível, é muito mais difícil determinar quem deve ir em seguida. Muitos protocolos para resolver o problema são conhecidos. Eles constituem o conteúdo deste capítulo ter. Na literatura, os canais de transmissão às vezes são chamados de **multiacesso canais** ou **canais de acesso aleatório**.

Os protocolos usados para determinar quem vai a seguir em um canal multiacesso serão longo para uma subcamada da camada de enlace chamada **MAC (Medium Access Control)** subcamada. A subcamada MAC é especialmente importante em LANs, particularmente sem fio porque sem fio é naturalmente um canal de transmissão. WANs, em contraste, use links ponto a ponto, exceto para redes de satélite. Porque multiacesso canais e LANs estão tão intimamente relacionados, neste capítulo discutiremos LANs em

257

Página 282

258

O SUBCAMADA DE CONTROLE DE ACESSO MÉDIO INDIVÍDUO. 4

geral, incluindo alguns problemas que não fazem parte estritamente da subcamada MAC, mas o assunto principal aqui será o controle do canal.

Tecnicamente, a subcamada MAC é a parte inferior da camada de enlace, então logicamente, deveríamos ter estudado antes de examinar todos os proto-pontos ponto a ponto cols no cap. 3. No entanto, para a maioria das pessoas, é mais fácil entender os protocolos envolvendo várias partes após os protocolos de duas partes serem bem compreendidos. Por isso razão pela qual nos desviamos ligeiramente de uma ordem estrita de apresentação de baixo para cima.

4.1 O PROBLEMA DE ALOCAÇÃO DE CANAIS

O tema central deste capítulo é como alocar um único canal de transmissão entre usuários concorrentes. O canal pode ser uma parte do espectro sem fio em uma região geográfica, ou um único fio ou fibra óptica para o qual vários nós estão conectados. Isso não importa. Em ambos os casos, o canal conecta cada usuário a todos os outros usuários e qualquer usuário que faz uso total do canal interfere com outros usuários que também desejam usar o canal.

Vamos primeiro olhar para as deficiências dos esquemas de alocação estática para rajadas tráfego. Em seguida, apresentaremos as principais premissas usadas para modelar a dinâmica esquemas que examinaremos nas seções a seguir.

4.1.1 Alocação de canal estático

A maneira tradicional de alocar um único canal, como um tronco de telefone, entre vários usuários concorrentes é cortar sua capacidade usando um dos esquemas de multiplexação que descrevemos na Seç. 2.5, como FDM (Divisão de Frequência Multiplexing). Se houver N usuários, a largura de banda é dividida em N de tamanhos iguais porções, com cada usuário sendo atribuído a uma porção. Uma vez que cada usuário tem um banda de frequência, agora não há interferência entre os usuários. Quando há apenas um número pequeno e constante de usuários, cada um com um fluxo constante ou intenso carga de tráfego, esta divisão é um mecanismo de alocação simples e eficiente. UMA um exemplo sem fio são as estações de rádio FM. Cada estação obtém uma parte da banda FM e usa-o na maioria das vezes para transmitir seu sinal.

No entanto, quando o número de remetentes é grande e variável ou o tráfego é intermitente, o FDM apresenta alguns problemas. Se o espectro for dividido em N regiões e menos de N usuários estão atualmente interessados em se comunicar, um grande pedaço de espectro valioso será desperdiçado. E se mais de N usuários quiserem se comunicar

cate, alguns deles terão permissão negada por falta de largura de banda, mesmo que alguns dos usuários a quem foi atribuída uma banda de frequência, dificilmente transmitem ou receve nada.

Mesmo supondo que o número de usuários pudesse de alguma forma ser mantido constante em N , dividir o único canal disponível em algum número de subcanais estáticos é

Página 283

SEC. 4,1

O PROBLEMA DE ALOCAÇÃO DE CANAIS

259

inerentemente ineficiente. O problema básico é que, quando alguns usuários estão quietos, sua largura de banda é simplesmente perdida. Eles não estão usando, e ninguém mais tem permissão para use-o também. Uma alocação estática é um ajuste inadequado para a maioria dos sistemas de computador, nos quais o tráfego de dados é extremamente intermitente, muitas vezes com pico de tráfego para taxas de tráfego de 1000: 1. Conseqüentemente, a maioria dos canais ficará ociosa na maior parte do tempo.

O fraco desempenho do FDM estático pode ser facilmente visto com uma simples fila cálculo da teoria ing. Vamos começar encontrando o tempo médio de atraso, T , para enviar um quadro em um canal de capacidade C bps. Assumimos que os frames chegam randomicamente com uma taxa média de chegada de λ frames / seg, e que os frames variam em comprimento com um comprimento médio de $1/\mu$ bits. Com esses parâmetros, a taxa de serviço do canal é μC frames / seg. Um resultado padrão da teoria das filas é

$$T =$$

$$\frac{1}{\mu C - \lambda}$$

1

(Para os curiosos, este resultado é para uma fila "M / M / 1". Requer que o domínio dos tempos entre a chegada dos quadros e os comprimentos dos quadros seguem uma distribuição exponencial, ou equivalentemente ser o resultado de um processo de Poisson.) Em nosso exemplo, se C é 100 Mbps, o comprimento médio do quadro, $1/\mu$, é 10.000 bits, e a taxa de chegada de quadros, λ , é de 5000 quadros / seg, então $T = 200 \mu\text{seg}$. Observe que se ignorarmos o atraso na fila e apenas perguntarmos quanto tempo leva para enviar um bit frame em uma rede de 100 Mbps, obteríamos a resposta (incorrecta) de 100 μsec . Esse resultado só é válido quando não há contenção para o canal.

Agora vamos dividir o único canal em N subcanais independentes, cada com capacidade C/N bps. A taxa média de entrada em cada um dos subcanais agora seja λ/N . Recomputando T , obtemos

$$T_N =$$

$$\frac{1}{\mu(C/N) - (\lambda/N)}$$

1

=

$$\frac{\mu C - \lambda}{N}$$

$$= NT$$

(4-1)

O atraso médio para o canal dividido é N vezes pior do que se todos os quadros estavam de alguma forma magicamente organizados em uma grande fila central. Este mesmo resultado

diz que um saguão de banco cheio de caixas eletrônicos é melhor ter uma única fila alimentando todas as máquinas do que uma fila separada na frente de cada máquina.

Precisamente os mesmos argumentos que se aplicam ao FDM também se aplicam a outras formas de dividindo estaticamente o canal. Se fôssemos usar multiplexação por divisão de tempo (TDM), e atribuir a cada utilizador todos os N° intervalo de tempo, se um utilizador não usa o alocar ranhura designada, ficaria apenas em pousio. O mesmo aconteceria se dividíssemos a rede trabalha fisicamente. Usando nosso exemplo anterior novamente, se fôssemos substituir a Rede de 100 Mbps com 10 redes de 10 Mbps cada e alocar estaticamente cada uma

usuário para um deles, o atraso médio saltaria de 200 μ s para 2 ms. Uma vez que nenhum dos métodos tradicionais de alocação de canais estáticos funcionam bem com tráfego em rajadas, agora exploraremos métodos dinâmicos.

260

O SUBCAMADA DE CONTROLE DE ACESSO MÉDIO INDIVÍDUO. 4

4.1.2 Premissas para alocação dinâmica de canais

Antes de chegarmos ao primeiro dos muitos métodos de alocação de canal neste capítulo Além disso, vale a pena formular cuidadosamente o problema de alocação. Subjacente a tudo o trabalho realizado nesta área são as seguintes cinco premissas principais:

1. **Tráfego independente**. O modelo consiste em N estações independentes (por exemplo, computadores, telefones), cada um com um programa ou usuário que gera quadros para transmissão. O número esperado de quadros gerados atado em um intervalo de comprimento Δt é $\lambda\Delta t$, onde λ é uma constante (o arritmo val de novos quadros). Uma vez que um quadro foi gerado, a estação é bloqueada e não faz nada até que o quadro tenha sido bem sucedido transmitido com sucesso.

2. **Canal único**. Um único canal está disponível para todas as comunicações. Todas as estações podem transmitir nele e todas podem receber dele. As estações são consideradas igualmente capazes, embora os protocolos possam atribuir-lhes funções diferentes (por exemplo, prioridades).

3. **Colisões observáveis**. Se dois quadros forem transmitidos simultaneamente obviamente, eles se sobrepõem no tempo e o sinal resultante é distorcido. Isto evento é chamado de **colisão**. Todas as estações podem detectar que uma colisão ocorreu. Um quadro colidido deve ser transmitido novamente mais tarde. Não erros diferentes daqueles gerados por colisões ocorrem.

4. **Tempo contínuo ou com fenda**. O tempo pode ser considerado contínuo, em qual transmissão de quadro de caso pode começar a qualquer momento. Alternativamente, o tempo pode ser dividido em intervalos discretos (chamados slots). As transmissões de quadros devem então começar no início de um slot. UMA slot pode conter 0, 1 ou mais frames, correspondendo a um slot inativo, uma transmissão bem-sucedida ou colisão, respectivamente.

5. **Sentido de operadora ou sem sentido de operadora**. Com o sentido de portador como-soma, as estações podem dizer se o canal está em uso antes de tentar usar isto. Nenhuma estação tentará usar o canal enquanto ele for detectado como ocupado. Se não houver detecção de portadora, as estações não podem detectar o canal antes de tentar usá-lo. Eles apenas vão em frente e transmitem. Só mais tarde eles podem determinar se a transmissão foi bem-sucedida.

Alguma discussão dessas suposições é necessária. O primeiro diz que as chegadas de quadros são independentes, tanto entre as estações quanto em uma estação específica, e

que os quadros são gerados de forma imprevisível, mas a taxa constante. Na verdade, isso assumption não é um modelo particularmente bom de tráfego de rede, como é bem conhecido que os pacotes vêm em rajadas em uma faixa de escalas de tempo (Paxson e Floyd, 1995; e Leland et al., 1994). No entanto, os **modelos de Poisson**, como são frequentemente chamados, são úteis porque são matematicamente tratáveis. Eles nos ajudam a analisar

SEC. 4,1

O PROBLEMA DE ALOCAÇÃO DE CANAIS

261

protocolos para entender aproximadamente como o desempenho muda ao longo de uma operação alcance e como ele se compara com outros designs.

A suposição de canal único é o coração do modelo. Sem meios externos para comunicar existir. As estações não podem levantar as mãos para solicitar que o professor chamá-los, então teremos que encontrar soluções melhores.

As três premissas restantes dependem da engenharia do sistema, e diremos quais suposições são válidas quando examinamos um protocolo específico. A suposição de colisão é básica. As estações precisam de alguma forma de detectar colisões se eles devem retransmitir quadros em vez de deixá-los se perder. Para canais com fio, o hardware do nó pode ser projetado para detectar colisões quando elas ocorrem. As estações podem então encerrar suas transmissões prematuramente para evitar o desperdício de capacidade. Esta detecção é muito mais difícil para canais sem fio, então as colisões são geralmente inferida após o fato pela falta de um quadro de reconhecimento esperado. Isto é também é possível que alguns quadros envolvidos em uma colisão sejam recebidos com sucesso, dependendo dos detalhes dos sinais e do hardware de recepção. No entanto, este situação não é o caso comum, então vamos supor que todos os quadros envolvidos em um colisão são perdidas. Também veremos protocolos que são projetados para evitar colisões ions ocorram em primeiro lugar.

A razão para as duas suposições alternativas sobre o tempo é que o tempo dividido pode ser usado para melhorar o desempenho. No entanto, requer que as estações sigam um mestre do relógio ou sincronizar suas ações entre si para dividir o tempo em dis-intervalos de crea. Portanto, nem sempre está disponível. Vamos discutir e analisar sistemas com ambos os tipos de tempo. Para um determinado sistema, apenas um deles é válido. Da mesma forma, uma rede pode ter sensor de portadora ou não. Redes com fio geralmente terá senso de portadora. As redes sem fio nem sempre podem usá-lo efetivamente porque nem todas as estações podem estar dentro do alcance de rádio de todas as outras ção. Da mesma forma, o sensor de portadora não estará disponível em outras configurações em que um sta-ção não pode se comunicar diretamente com outras estações, por exemplo, um modem a cabo em que as estações devem se comunicar através do headend do cabo. Observe que a palavra "portadora ", neste sentido, refere-se a um sinal no canal e não tem nada a ver com as operadoras comuns (por exemplo, empresas de telefonia) que datam dos dias da Pony Express.

Para evitar qualquer mal-entendido, é importante notar que nenhum protocolo de multiacesso col garante entrega confiável. Mesmo na ausência de colisões, o receptor pode ter copiado parte do quadro incorretamente por vários motivos. Outras partes de a camada de link ou camadas superiores fornecem confiabilidade.

4.2 PROTOCOLOS DE ACESSO MÚLTIPLOS

Muitos algoritmos para alocar um canal de acesso múltiplo são conhecidos. No nas seções seguintes, estudaremos uma pequena amostra das mais interessantes e dê alguns exemplos de como eles são comumente usados na prática.

Página 286

262

O SUBCAMADA DE CONTROLE DE ACESSO MÉDIO
INDIVÍDUO. 4

4.2.1 ALOHA

A história do nosso primeiro MAC começa no primitivo Havaí no início dos anos 1970. No neste caso, "primitivo " pode ser interpretado como "não ter um sistema telefônico funcionando tem. " Isso não tornou a vida mais agradável para o pesquisador Norman Abramson e seus colegas da Universidade do Havaí que estavam tentando conectar usuários em re-ilhas mote para o computador principal em Honolulu. Enfiando seus próprios cabos sob o Oceano Pacífico não estava nas cartas, então eles procuraram uma solução diferente. O que eles encontraram usava rádios de curto alcance, com cada terminal de usuário compartilhando a mesma frequência upstream para enviar quadros para o computador central. Incluía um método simples e elegante para resolver o problema de alocação de canais. Trabalho deles foi ampliado por muitos pesquisadores desde então (Schwartz e Abramson, 2009). Embora o trabalho de Abramson, chamado de sistema ALOHA, usasse radiodifusão de rádio, a ideia básica é aplicável a qualquer sistema em que

usuários descoordenados estão competindo pelo uso de um único canal compartilhado. Discutiremos duas versões de ALOHA aqui: puro e com slot. Eles diferem com relação a se o tempo é contínuo, como na versão pura, ou dividido em slots discretos nos quais todos os quadros devem se encaixar.

ALOHA puro

A ideia básica de um sistema ALOHA é simples: permitir que os usuários transmitam sempre eles têm dados a serem enviados. Haverá colisões, é claro, e a colisão os quadros serão danificados. Os remetentes precisam de alguma forma para descobrir se é esse o caso. No

o sistema ALOHA, após cada estação enviar seu quadro para o computador central, este computador retransmite o quadro para todas as estações. Uma estação emissora pode portanto, ouça a transmissão do hub para ver se o quadro foi transmitido. No outros sistemas, como LANs com fio, o remetente pode ser capaz de ouvir colisões ions durante a transmissão.

Se o quadro foi destruído, o remetente apenas espera um período de tempo aleatório e envia novamente. O tempo de espera deve ser aleatório ou os mesmos frames irão colidir repetidamente, em sincronia. Sistemas em que vários usuários compartilham um mesmo canal de uma forma que pode levar a conflitos são conhecidos como sistemas de **contenção**. Um esboço da geração de quadros em um sistema ALOHA é dado na Figura 4-1. Nós fizeram todos os quadros com o mesmo comprimento porque a taxa de transferência do sistema ALOHA

tem é maximizado por ter um tamanho de quadro uniforme, em vez de permitir vários quadros de comprimento adequado.

Sempre que dois frames tentarem ocupar o canal ao mesmo tempo, haverá ser uma colisão (como visto na Fig. 4-1) e ambos ficarão truncados. Se o primeiro pedaço de o novo quadro se sobrepõe apenas ao último pedaço de um quadro que está quase terminado, ambos os quadros serão totalmente destruídos (ou seja, terão somas de verificação incorretas) e ambos serão tem que ser retransmitido mais tarde. A soma de verificação não (e não deve) distinguir palpitar entre uma perda total e um quase acidente. Ruim é ruim.

Página 287

SEC. 4,2 PROTÓCOLOS DE ACESSO MÚLTIPLOS

263

Do utilizador
UMA
B
C
D
E
Tempo
Colisão
Colisão

Figura 4-1. No ALOHA puro, os quadros são transmitidos em momentos completamente arbitrários. Uma questão interessante é: qual é a eficiência de um canal ALOHA? No outras palavras, que fração de todos os quadros transmitidos escapa das colisões sob estes circunstâncias caóticas? Vamos primeiro considerar uma coleção infinita de usuários digitando em seus terminais (estações). Um usuário está sempre em um de dois estados: digitando ou aguardando. Inicialmente, todos os usuários estão no estado de digitação. Quando uma linha é finalizada, o usuário para de digitar, esperando uma resposta. A estação então transmite um quadro de con manter a linha sobre o canal compartilhado para o computador central e verificar o canal para ver se foi bem sucedido. Nesse caso, o usuário vê a resposta e volta para digitando. Caso contrário, o usuário continua a esperar enquanto a estação retransmite o quadro repetidamente até que seja enviado com sucesso. Deixe o "tempo de quadro" denotar a quantidade de tempo necessária para transmitir o padrão quadro padrão, de comprimento fixo (ou seja, o comprimento do quadro dividido pela taxa de bits). Neste ponto, assumimos que os novos quadros gerados pelas estações são bem modelados

por uma distribuição de Poisson com uma média de N quadros por tempo de quadro. (O infinito-a suposição da população é necessária para garantir que o N não diminua conforme os usuários vir bloqueado.) Se $N > 1$, a comunidade de usuários está gerando quadros em um nível superior taxa que o canal pode suportar, e quase todos os quadros sofrerão uma colisão.

Para uma taxa de transferência razoável, esperaríamos $0 < N < 1$.

Além dos novos frames, as estações também geram retransmissões de frames que anteriormente sofreram colisões. Vamos ainda supor que o antigo e novos quadros combinados são bem modelados por uma distribuição de Poisson, com média de G quadros por tempo de quadro. Claramente, $G \geq N$. Em carga baixa (ou seja, $N \sim 0$), haverá poucas colisões, portanto, poucas retransmissões, então $G \sim N$. Em alta carga, haverá muitas colisões, então $G > N$. Sob todas as cargas, o rendimento, S , é apenas o oferecido carga, G , vezes a probabilidade, P_0 , de uma transmissão bem-sucedida - isto é,

$$S = GP_0, \text{ onde } P_0 \text{ é a probabilidade de que um quadro não sofra uma colisão.}$$

Um quadro não sofrerá uma colisão se nenhum outro quadro for enviado dentro de um frame time de seu início, conforme mostrado na Figura 4-2. Sob quais condições o

Página 288

264

O SUBCAMADA DE CONTROLE DE ACESSO MÉDIO INDIVÍDUO. 4

a moldura sombreada chega sem danos? Seja t o tempo necessário para enviar um quadro. E se qualquer outro usuário gerou um quadro entre o tempo t_0 e $t_0 + t$, o final desse o quadro colidirá com o início do sombreado. Na verdade, o sombreado o destino do frame já estava selado antes mesmo do primeiro bit ser enviado, mas desde ALOHA uma estação não escuta o canal antes de transmitir, não tem como de saber que outro quadro já estava em andamento. Da mesma forma, qualquer outro quadro iniciado entre $t_0 + t$ e $t_0 + 2t$ colidirá com o final do quadro sombreado.

Colide com
o começo de
o sombreado
quadro, Armação
Colide com
o fim de
o sombreado
quadro, Armação
 t
 t_0
 $t_0 + t$
 $t_0 + 2t$
Tempo $t_0 + 3t$
Vulnerável

Figura 4-2. Período vulnerável para a moldura sombreada.

A probabilidade de que k quadros sejam gerados durante um determinado tempo de quadro, em quais quadros G são esperados, é dado pela distribuição de Poisson

$$\Pr [k] = \frac{k^k}{k!} e^{-G} \quad (4-2)$$

então a probabilidade de zero frames é apenas e^{-G} . Em um intervalo de tempo de dois quadros, o número médio de quadros gerados é $2G$. A probabilidade de nenhum quadro ser iniciado durante todo o período vulnerável é, portanto, dado por $P_0 = e^{-2G}$.

Usando
 $S = GP_0$, obtemos
 $S = Ge^{-2G}$

A relação entre o tráfego oferecido e a taxa de transferência é mostrada em Fig. 4-3. A taxa de transferência máxima ocorre em $L = 0,5$, com $S = 1/2 e^{-2G}$, o que é cerca de 0,184. Em outras palavras, o melhor que podemos esperar é a utilização do canal de 18%. Esse resultado não é muito animador, mas com todos transmitindo à vontade, dificilmente poderíamos esperar uma taxa de sucesso de 100%.

ALOHA com fenda

Logo depois que ALOHA entrou em cena, Roberts (1972) publicou um método para dobrar a capacidade de um sistema ALOHA. Sua proposta era dividir o tempo em intervalos discretos chamados **slots**, cada intervalo correspondendo a um quadro. Isto

SEC. 4,2
PROTOCOLOS DE ACESSO MÚLTIPLOS

265

0,40
0,30
0,20
0,10
0
0,5
1,0
1,5
G (tentativas por tempo de pacote)
2,0
3,0
S
(Taxa de transferência
por
quadro, Armação
Tempo)
ALOHA com fenda: $S = Ge^{-G}$
ALOHA puro: $S = Ge^{-2G}$

Figura 4-3. Taxa de transferência versus tráfego oferecido para sistemas ALOHA.
abordagem requer que os usuários concordem com os limites dos slots. Uma maneira de obter sincronização seria ter uma estação especial emitindo um pip no início de cada intervalo, como um relógio.

No método de Roberts, que passou a ser conhecido como **slotted ALOHA** - em contraste com o **ALOHA puro** de Abramson - uma estação não tem permissão para enviar quando-sempre o usuário digita uma linha. Em vez disso, é necessário aguardar o início do próximo slot. Assim, o tempo contínuo ALOHA é transformado em um tempo discreto. Isso reduz pela metade o período de vulnerabilidade. Para ver isso, observe a Fig. 4-3 e imagine o colisões que agora são possíveis. A probabilidade de nenhum outro tráfego durante o mesmo slot que nosso quadro de teste é e^{-G} ,

e^{-G} , que leva a

$$S = Ge^{-G}$$

e^{-G}

(4-3)

Como você pode ver na Fig. 4-3, os picos de ALOHA ranhurados em $G = 1$, com um rendimento de $S = 1/e$ ou cerca de 0,368, o dobro do ALOHA puro. Se o sistema estiver operando em $G = 1$, a probabilidade de um slot vazio é 0,368 (da Eq. 4-2). O melhor que nós podemos esperar para usar o ALOHA com slot em 37% dos slots vazios, 37% de sucessos e 26% de colisões. Operar com valores mais altos de G reduz o número de vazios mas aumenta o número de colisões exponencialmente. Para ver como esta rapidez crescente de colisões com G acontece, considere a transmissão de um teste

quadro, Armação. A probabilidade de evitar uma colisão é e^{-G} ,

e^{-G} , que é a probabilidade-

É necessário que todas as outras estações fiquem em silêncio nesse slot. A probabilidade de uma colisão é

então apenas $1 - e^{-G}$

e^{-G} . A probabilidade de uma transmissão exigir exatamente k tentativas (ou seja, $k - 1$ colisões seguidas por um sucesso) é

$$P_k = e^{-G} (1 - e^{-G})^{k-1}$$

$$(1 - e^{-G})^{k-1}$$

O número esperado de transmissões, E , por linha digitada em um terminal é então

$$E =$$

$$\sum_{k=1}^{\infty}$$

$$\infty$$

$$kP_k = \sum_{k=1}^{\infty} ke^{-G}(1-e^{-G})^{k-1} = e^{-G}$$

266

O SUBCAMADA DE CONTROLE DE ACESSO MÉDIO INDIVÍDUO. 4

Como resultado da dependência exponencial de E sobre G , pequenos aumentos no a carga do canal pode reduzir drasticamente seu desempenho.

ALOHA com fenda é notável por uma razão que pode não ser inicialmente óbvia. isto foi criado na década de 1970, usado em alguns sistemas experimentais iniciais, então quase esquecido. Quando o acesso à Internet por cabo foi inventado, de repente era um problema de como alocar um canal compartilhado entre vários concorrentes Comercial. O ALOHA com fenda foi retirado da lata de lixo para salvar o dia. Mais tarde, ter várias etiquetas RFID conversando com o mesmo leitor RFID apresentou outra variação sobre o mesmo problema. ALOHA com fenda, com uma pitada de outras ideias misturadas, novamente veio para o resgate. Muitas vezes acontece que protocolos que são perfeitamente válido cair em desuso por razões políticas (por exemplo, alguma grande empresa quer todos para fazer as coisas do seu jeito) ou devido às tendências de tecnologia em constante mudança. Então, anos depois

alguma pessoa inteligente percebe que um protocolo há muito descartado resolve seus problemas atuais

Iem. Por esta razão, neste capítulo estudaremos uma série de protocolos elegantes que não estão atualmente em uso generalizado, mas podem ser facilmente usados em aplicações futuras

, desde que um número suficiente de designers de rede as conheça. Claro, nós também estudará muitos protocolos que também estão em uso.

4.2.2 Protocolos de Acesso Múltiplo do Carrier Sense

Com o ALOHA com slot, a melhor utilização de canal que pode ser alcançada é $1/e$.

Este baixo resultado dificilmente é surpreendente, uma vez que, com estações transmitindo à vontade, com-

sabendo o que as outras estações estão fazendo, deve haver muitos colisões ions. Em LANs, no entanto, muitas vezes é possível para as estações detectar quais outras estações ações estão fazendo e, portanto, adaptar seu comportamento de acordo. Essas redes podem alcançar uma utilização muito melhor do que $1/e$. Nesta seção, discutiremos alguns protocolos para melhorar o desempenho.

Protocolos nos quais as estações ouvem uma portadora (ou seja, uma transmissão) e atuam consequentemente, são chamados de **protocolos de detecção de portadora**. Vários deles foram propostas e foram analisadas em detalhes há muito tempo. Por exemplo, veja Kleinrock e Tobagi (1975). Abaixo, veremos várias versões do protótipo de detecção de portadora cols.

CSMA Persistente e Não Persistente

O primeiro protocolo de detecção de portadora que estudaremos aqui é chamado **1-persistent CSMA** (**Carrier Sense Multiple Access**). Isso é um pouco complicado para o sim-esquema mais simples de CSMA. Quando uma estação tem dados para enviar, ela primeiro escuta o canal

para ver se mais alguém está transmitindo naquele momento. Se o canal estiver ocioso, o estação envia seus dados. Caso contrário, se o canal estiver ocupado, a estação apenas espera até que fique ocioso. Em seguida, a estação transmite um quadro. Se ocorrer uma colisão, o

estação espera um período de tempo aleatório e começa tudo de novo. O protocolo é chamado 1-persistente porque a estação transmite com uma probabilidade de 1 quando encontra o canal inativo.

Você pode esperar que este esquema evite colisões, exceto no caso raro de envios simultâneos, mas na verdade não é assim. Se duas estações ficarem prontas em no meio da transmissão de uma terceira estação, ambos esperarão educadamente até a transmissão termina, e então ambos começarão a transmitir exatamente simultaneamente, resultando em uma colisão. Se eles não fossem tão impacientes, haveria menos colisões íons.

Mais sutilmente, o retardo de propagação tem um efeito importante nas colisões. Há uma chance de que, logo após uma estação começar a enviar, outra estação será-pronta para enviar e sentir o canal. Se o sinal da primeira estação ainda não alcançou o segundo, o último detectará um canal ocioso e também começará envio, resultando em uma colisão. Essa chance depende do número de quadros que cabem no canal ou o **produto de atraso de largura de banda** do canal. Se apenas um uma pequena fração de um quadro se encaixa no canal, o que é o caso na maioria das LANs desde o atraso de propagação é pequeno, a chance de uma colisão acontecer é pequena. O quanto maior o produto de atraso de largura de banda, mais importante se torna esse efeito, e pior é o desempenho do protocolo.

Mesmo assim, este protocolo tem melhor desempenho do que o ALOHA puro porque ambos as estações têm a decência de desistir de interferir no quadro da terceira estação.

Exatamente o mesmo se aplica ao ALOHA com fenda.

Um segundo protocolo de detecção de portadora é o **CSMA não persistente**. Neste protocolo, um tentativa consciente é feita para ser menos gananciosa do que na anterior. Como antes, um estação detecta o canal quando deseja enviar um quadro, e se ninguém mais está envio, a própria estação começa a fazê-lo. No entanto, se o canal já estiver em uso, a estação não sente continuamente com o propósito de apreendê-lo imediatamente após detectar o fim da transmissão anterior. Em vez disso, ele espera um período aleatório de tempo e, em seguida, repete o algoritmo. Consequentemente, este algoritmo ritmo leva a melhor utilização do canal, mas atrasos mais longos do que 1-persistent CSMA.

O último protocolo é o **CSMA p-persistente**. Aplica-se a canais com slot e funciona da seguinte maneira. Quando uma estação fica pronta para enviar, ela detecta o canal. E se está ocioso, ele transmite com uma probabilidade p . Com uma probabilidade $q = 1 - p$, ele difere até o próximo slot. Se esse slot também estiver ocioso, ele transmite ou adia novamente, com probabilidades p e q . Este processo é repetido até que o quadro tenha sido transmitido ou outra estação começou a transmitir. No último caso, o estação azarada age como se tivesse ocorrido uma colisão (ou seja, espera um tempo aleatório e começa novamente). Se a estação inicialmente sentir que o canal está ocupado, ela espera até o próximo slot e aplica o algoritmo acima. IEEE 802.11 usa um refinamento de p-persistent CSMA que discutiremos na Seç. 4,4.

A Figura 4-4 mostra a taxa de transferência computada versus tráfego oferecido para todos os três protocolos, bem como para ALOHA puro e com fenda.

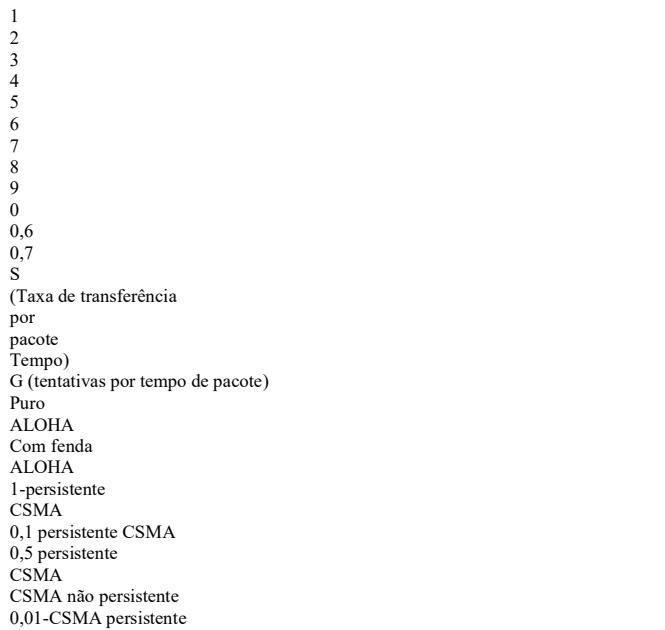


Figura 4-4. Comparação da utilização do canal versus carga para vários protocolos de acesso dom.

CSMA com detecção de colisão

Protocolos CSMA persistentes e não persistentes são definitivamente uma melhoria sobre ALOHA porque eles garantem que nenhuma estação comece a transmitir enquanto o canal está ocupado. No entanto, se duas estações sentirem que o canal está ocioso e começar transmitindo simultaneamente, seus sinais ainda irão colidir. Outra melhoria é para as estações detectarem rapidamente a colisão e pararem de transmitir abruptamente, (em vez de finalizá-los), pois, de qualquer maneira, eles estão irremediavelmente truncados. Isto estratégia economiza tempo e largura de banda.

Este protocolo, conhecido como **CSMA / CD** (**CSMA com detecção de colisão**), é a base da LAN Ethernet clássica, por isso vale a pena dedicar algum tempo para olhar em detalhes. É importante perceber que a detecção de colisão é um procedimento analógico ess. O hardware da estação deve ouvir o canal durante a transmissão. E se o sinal que ele lê de volta é diferente do sinal que está emitindo, ele sabe que um colisão está ocorrendo. As implicações são que um sinal recebido não deve ser minúsculo em comparação com o sinal transmitido (o que é difícil para wireless, pois o sinal recebido nais podem ser 1.000.000 vezes mais fracos do que os sinais transmitidos) e que o módulo ção deve ser escolhida para permitir que colisões sejam detectadas (por exemplo, uma colisão de dois 0-

sinais de volt podem ser impossíveis de detectar).

CSMA / CD, bem como muitos outros protocolos de LAN, usa o modelo conceitual da Fig. 4-5. No ponto marcado t_0 , uma estação terminou de transmitir seu quadro. Qualquer outra estação que tenha um quadro a enviar pode agora tentar fazê-lo. Se dois ou mais estações decidirem transmitir simultaneamente, haverá uma colisão. Se uma estação detecta uma colisão, aborta sua transmissão, espera um período aleatório de tempo, e, em seguida, tenta novamente (assumindo que nenhuma outra estação começou a transmitir no

entretanto). Portanto, nosso modelo para CSMA / CD consistirá em alternar con períodos de atenção e transmissão, com períodos ociosos ocorrendo quando todas as estações estão quieto (por exemplo, por falta de trabalho).

período
Ocioso
período
t₀
Quadro, Armação
Quadro, Armação
Quadro, Armação
Quadro, Armação
Tempo

Figura 4-5. O CSMA / CD pode estar em contenção, transmissão ou estado ocioso.

Agora, vejamos os detalhes do algoritmo de contenção. Suponha que duas ambas as estações começam a transmitir exatamente no tempo t_0 . Quanto tempo vai demorar para eles

perceber que eles colidiram? A resposta é vital para determinar a duração do o período de contenção e, portanto, qual será o atraso e a taxa de transferência. O tempo mínimo para detectar a colisão é apenas o tempo que o sinal leva para propagar de uma estação para a outra. Com base nessas informações, você pode pensar que uma estação que não ouviu uma colisão por um tempo igual ao cabo completo o tempo de propagação após o início de sua transmissão pode ter certeza de que agarrou o cabo. Por "apreendido", queremos dizer que todas as outras estações sabem que está transmitindo e não interferir. Esta conclusão está errada.

Considere o seguinte cenário de pior caso. Deixe o tempo para um sinal para propagar entre as duas estações mais distantes seja τ . Em t_0 , uma estação começa a transmitting. Em $t_0 + \tau - \epsilon$, um instante antes de o sinal chegar ao estado mais distante a estação também começa a transmitir. Claro, ele detecta a colisão al-mais instantaneamente e para, mas o pequeno ruído causado pela colisão não volte para a estação original até o tempo $2\tau - \epsilon$. Em outras palavras, no pior caso uma estação não pode ter certeza de que ocupou o canal até que tenha transmitido por 2τ sem ouvir uma colisão.

Com esse entendimento, podemos pensar na contenção CSMA / CD como um slot Sistema ALOHA com largura de ranhura de 2τ . Em um cabo coaxial de 1 km de comprimento, $\tau \sim 5 \mu\text{sec}$. A diferença para CSMA / CD em comparação com ALOHA com fenda é que slots em que apenas uma estação transmite (ou seja, em que o canal é apreendido) são seguidos pelo resto de um quadro. Essa diferença vai melhorar muito o desempenho se o tempo de quadro for muito maior do que o tempo de propagação.

4.2.3 Protocolos livres de colisão

Embora as colisões não ocorram com CSMA / CD, uma vez que uma estação tem unambigamente capturados visivelmente no canal, elas ainda podem ocorrer durante o período de contenção. Essas colisões afetam negativamente o desempenho do sistema, especialmente quando o

Página 294

270

O SUBCAMADA DE CONTROLE DE ACESSO MÉDIO INDIVÍDUO. 4

produto de atraso de largura de banda é grande, como quando o cabo é longo (ou seja, grande τ) e os quadros são curtos. Não só as colisões reduzem a largura de banda, mas também tornam hora de enviar uma variável de quadro, que não é um bom ajuste para tráfego em tempo real, como voz sobre IP. CSMA / CD também não é universalmente aplicável.

Nesta seção, examinaremos alguns protocolos que resolvem a contenção de o canal sem nenhuma colisão, nem mesmo durante o período de contenção.

A maioria desses protocolos não são usados atualmente nos sistemas principais, mas em um rápido campo em mudança, tendo alguns protocolos com excelentes propriedades disponíveis para sistemas de segurança costumam ser uma coisa boa.

Nos protocolos a serem descritos, assumimos que existem exatamente N estações, cada um programado com um endereço único de 0 a $N - 1$. Não importa que algumas estações podem ficar inativas parte do tempo. Também assumimos que a propagação o atraso é insignificante. A questão básica permanece: qual estação obtém o canal após uma transmissão bem-sucedida? Continuamos usando o modelo da Fig. 4-5 com seu slots de contenção discretos.

Um protocolo de bitmap

Em nosso primeiro protocolo livre de colisão, o **método de mapa de bits básico**, cada período de atenção consiste em exatamente N slots. Se a estação 0 tiver um quadro para enviar, ela transmite 1 bit durante o slot 0. Nenhuma outra estação pode transmitir durante este slot. Independentemente do que a estação 0 faz, a estação 1 tem a oportunidade de transmitir um 1 bit durante o slot 1, mas apenas se houver um quadro na fila. Em geral, a estação j pode anunciar que tem um quadro para enviar inserindo um bit 1 no slot j . Afinal N slots passaram, cada estação tem conhecimento completo de quais estações desejam transmitir. Nesse ponto, eles começam a transmitir quadros em ordem numérica (ver Fig. 4-6).

```

0 1
1
1
1
1
1
1
5
1
1
3
7
2 3 4 5 6 7
0 1 2 3 4 5 6 7
0 1 2 3 4 5 6 7
1
8 slots de contenção
Molduras
8 slots de contenção
2
d

```

Figura 4-6. O protocolo básico de bitmap.

Como todos concordam em quem vai em seguida, nunca haverá colisões.

Depois que a última estação pronta transmitiu seu quadro, um evento que todas as estações podem facilitar

Monitorar, outro período de contenção de N bits é iniciado. Se uma estação ficar pronta logo após o seu slot de bit ter passado, ele está sem sorte e deve permanecer em silêncio até cada estação teve uma chance e o mapa de bits voltou ao normal.

Protocolos como este em que o desejo de transmitir é transmitido antes do acionamento real são chamados de **protocolos de reserva** porque reservam canal propriedade com antecedência e evitar colisões. Vamos analisar brevemente o desempenho deste protocolo. Por conveniência, mediremos o tempo em unidades do slot de bit de contenção, com frames de dados consistindo em d unidades de tempo. Sob condições de baixa carga, o bitmap será simplesmente repetido várias vezes acabou, por falta de frames de dados. Considere a situação do ponto de vista de um estação com número baixo, como 0 ou 1. Normalmente, quando fica pronta para enviar, o slot "atual" estará em algum lugar no meio do mapa de bits. Na média, a estação terá que esperar $N/2$ slots para que a varredura atual termine e outro N slots completos para a seguinte varredura ser executada até a conclusão antes de começar a transmitting.

As perspectivas para estações de grande número são mais brilhantes. Geralmente, estes irão só tem que esperar meia varredura ($N/2$ slots de bits) antes de começar a transmitir. Altas estações numeradas raramente precisam esperar pela próxima varredura. Uma vez que estas estações devem esperar em média $1,5 N$ slots e as estações com numeração alta devem esperar em média $0,5 N$ slots, a média para todas as estações é N slots.

A eficiência do canal em baixa carga é fácil de calcular. A sobrecarga por quadro é N bits e a quantidade de dados é d bits, para uma eficiência de $d/(d+N)$.

Em alta carga, quando todas as estações têm algo para enviar o tempo todo, o N -período de contenção de bits é rateado sobre N frames, produzindo um overhead de apenas 1 bit

por quadro, ou uma eficiência de $d / (d + 1)$. O atraso médio para um quadro é igual a a soma do tempo que ele fica na fila dentro de sua estação, mais um adicional $(N - 1)d + N$ assim que chegar ao topo de sua fila interna. Este intervalo é quanto tempo leva para espere que todas as outras estações tenham sua vez de enviar um quadro e outro bitmap.

Passagem de Token

A essência do protocolo bit-map é que ele permite que cada estação transmita um quadro por sua vez em uma ordem predefinida. Outra maneira de fazer a mesma coisa é para passar uma pequena mensagem chamada **token** de uma estação para a próxima no mesmo ordem predefinida. O token representa a permissão para enviar. Se uma estação tem um quadro na fila para transmissão quando receber o token, ele pode enviar esse quadro antes de passar o token para a próxima estação. Se não tiver nenhum quadro enfileirado, ele simplesmente passa o token.

Em um protocolo **token ring**, a topologia da rede é usada para definir o ordem em que as estações enviam. As estações são conectadas uma à outra em um único anel. Passar o token para a próxima estação consiste simplesmente em receber o token em uma direção e transmitindo-o na outra direção, como visto em Fig. 4-7. Os quadros também são transmitidos na direção do token. Assim eles circularão ao redor do anel e alcançará qualquer estação que seja o destino. Como sempre, para parar o quadro de circular indefinidamente (como o token), alguma estação precisa

Página 296

272

O SUBCAMADA DE CONTROLE DE ACESSO MÉDIO INDIVÍDUO. 4

para removê-lo do anel. Esta estação pode ser a que enviou originalmente o quadro, após ter passado por um ciclo completo, ou a estação que era o destinatário pretendido do quadro.

Direção de
transmissão
Símbolo
Estação

Figura 4-7. Token ring.

Observe que não precisamos de um anel físico para implementar a passagem de token. O canal que conecta as estações pode ser um único barramento longo. Cada estação em seguida, usa o barramento para enviar o token para a próxima estação na sequência predefinida. A posse do token permite que uma estação use o barramento para enviar um quadro, como diante. Esse protocolo é chamado de **barramento de token**.

O desempenho da passagem de token é semelhante ao do protocolo de mapa de bits, embora os slots de contenção e os quadros de um ciclo estejam agora misturados. Depois de enviando um quadro, cada estação deve esperar por todas as N estações (incluindo ela mesma) para enviar

o token para seus vizinhos e as outras estações $N - 1$ para enviar um quadro, se eles tem um. Uma diferença sutil é que, uma vez que todas as posições no ciclo são equivalentes emprestado, não há preconceito para estações de número baixo ou alto. Para token ring, cada estação também está enviando o token apenas até a estação vizinha antes do protocolo dá o próximo passo. Cada token não precisa se propagar para todas as estações antes que o protocolo avance para a próxima etapa.

Os token ring surgiram como protocolos MAC com alguma consistência. A protocolo inicial de token ring (denominado "Token Ring" e padronizado como IEEE 802.5) era popular na década de 1980 como uma alternativa à Ethernet clássica. Na década de 1990, um token ring muito mais rápido chamado **FDDI** (**Fiber Distributed Data Interface**) foi derrotado por Ethernet comutada. Na década de 2000, um token ring chamado **RPR** (**Resilient Packet Ring**) foi definido como IEEE 802.17 para padronizar a mistura de metro-anéis de área politan em uso por ISPs. Nós nos perguntamos o que a década de 2010 terá a oferecer.

Contagem regressiva binária

Um problema com o protocolo de mapa de bits básico, e por extensão de passagem de token, é que a sobrecarga é de 1 bit por estação, por isso não se adapta bem a redes com

milhares de estações. Podemos fazer melhor do que isso usando estação binária vestidos com um canal que combina transmissões. Uma estação querendo usar o

Página 297

SEC. 4,2

PROTÓCOLOS DE ACESSO MÚLTIPLOS

273

canal agora transmite seu endereço como uma string de bits binários, começando com o alto pedido bit. Todos os endereços devem ter o mesmo comprimento. Os bits em cada anúncio posição de vestido de diferentes estações são BOOLEAN OU montadas pelo channel quando são enviados ao mesmo tempo. Chamaremos esse protocolo de **contagem binária para baixo**. Foi usado no Datakit (Fraser, 1987). Isso implica implicitamente que o transatrasos de missão são insignificantes, de modo que todas as estações veem bits declarados essencialmente em instantaneamente.

Para evitar conflitos, uma regra de arbitragem deve ser aplicada: assim que uma estação vê que uma posição de bit de alta ordem que é 0 em seu endereço foi sobrescrita com a 1, desiste. Por exemplo, se as estações 0010, 0100, 1001 e 1010 estiverem todas tentando para obter o canal, no primeiro tempo de bit as estações transmitem 0, 0, 1 e 1, respectivamente. Estes são combinados para formar um 1. Estações 0010 e 0100 consulte o 1 e sabem que uma estação de número mais alto está competindo pelo canal, então eles desista para a rodada atual. As estações 1001 e 1010 continuam.

O próximo bit é 0 e ambas as estações continuam. O próximo bit é 1, então estação 1001 desistir. O vencedor é a estação 1010 porque tem o endereço mais alto. Depois de vencendo a licitação, pode agora transmitir um frame, após o qual outra licitação o ciclo começa. O protocolo é ilustrado na Figura 4-8. Tem a propriedade de que estações numeradas têm uma prioridade mais alta do que estações numeradas mais baixas, que pode ser bom ou ruim, dependendo do contexto.

```
0 0 1 0  
0 - - -  
0 1 2 3  
Pouco tempo  
0 1 0 0  
0 - - -  
1 0 0 1  
1 0 0 -  
1 0 1 0  
1 0 1 0  
1 0 1 0  
Resultado  
Estações 0010  
e 0100 veja isso  
1 e desistir  
Estação 1001  
vê este 1  
e desiste
```

Figura 4-8. O protocolo binário de contagem regressiva. Um traço indica silêncio.

A eficiência do canal deste método é $d / (d + \log_2 N)$. Se, no entanto, o formato de quadro foi habilmente escolhido para que o endereço do remetente seja o primeiro campo no quadro, mesmo esses $\log_2 N$ bits não são perdidos e a eficiência é de 100%. A contagem regressiva binária é um exemplo de protocolo simples, elegante e eficiente que está esperando para ser redescoberto. Com sorte, ele encontrará um novo lar algum dia.

Página 298

274

O SUBCAMADA DE CONTROLE DE ACESSO MÉDIO INDIVÍDUO. 4

4.2.4 Protocolos de Contenção Limitada

Agora consideramos duas estratégias básicas para aquisição de canal em um rede de transmissão: contenção, como em CSMA, e protocolos sem colisão. Cada estratégia pode ser avaliada quanto ao quanto bem ela se sai em relação às duas importantes medidas de desempenho, retardo em baixa carga e eficiência do canal em alta carga. Sob

condições de carga leve, contenção (isto é, ALOHA puro ou com fenda) é preferível devido ao seu baixo atraso (já que as colisões são raras). Conforme a carga aumenta, a contenção torna-se cada vez menos atraente porque a sobrecarga associada ao canal a arbitragem se torna maior. O inverso é verdadeiro para o proto-protótipo livre de colisão cols. Em baixa carga, eles têm um atraso relativamente alto, mas à medida que a carga aumenta, o a eficiência do canal melhora (já que os overheads são fixos).

Obviamente, seria bom se pudéssemos combinar as melhores propriedades dos protocolos de contenção e livre de colisão, chegando a um novo protocolo que usava tensão em carga baixa para fornecer baixo atraso, mas usou uma técnica livre de colisão em carga alta para fornecer boa eficiência de canal. Esses protocolos, que chamaremos **protocolos de contenção limitada**, de fato existem, e irão concluir nosso estudo de redes de sentido superior.

Até agora, os únicos protocolos de contenção que estudamos foram simétricos ric. Ou seja, cada estação tenta adquirir o canal com alguma probabilidade, p , com todas as estações usando o mesmo p . Curiosamente, o sistema geral per-o desempenho às vezes pode ser melhorado usando um protocolo que atribui diferentes probabilidades para diferentes estações.

Antes de examinar os protocolos assimétricos, vamos revisar rapidamente o per-execução do caso simétrico. Suponha que k estações estão competindo por canal acesso nel. Cada um tem uma probabilidade p de transmissão durante cada slot. o probabilidade de que alguma estação adquira com sucesso o canal durante um determinado intervalo é a probabilidade de que qualquer estação transmita, com probabilidade p , e todas as outras $k - 1$ estações diferem, cada uma com probabilidade $1 - p$. Este valor é $kp(1 - p)^{k-1}$.

Para

encontrar o valor ótimo de p , diferenciamos em relação a p , definimos o resultado para zero e resolva para p . Fazendo isso, descobrimos que o melhor valor de p é $1/k$. Substituting $p = 1/k$, obtemos

$\Pr[\text{sucesso com } p \text{ ideal}] =$

k
 $k - 1$
 \vdots
 $k - 1$
 $(4-4)$

Essa probabilidade está representada na Figura 4-9. Para um pequeno número de estações, as chances

de sucesso são bons, mas assim que o número de estações chega a cinco, o a probabilidade caiu perto de seu valor assintótico de $1/e$.

Da Fig. 4-9, é bastante óbvio que a probabilidade de alguma aquisição de estação- o canal pode ser aumentado apenas diminuindo a quantidade de competição.

Os protocolos de contenção limitada fazem exatamente isso. Eles primeiro dividem as estações em grupos (não necessariamente separados). Apenas os membros do grupo 0 são permitidos

0
Probabilidade
do
sucesso
Número de estações prontas

Figura 4-9. Probabilidade de aquisição para um canal de contenção simétrico. para competir pelo slot 0. Se um deles tiver sucesso, ele adquire o canal e transmits sua moldura. Se a fenda estiver em pousio ou se houver uma colisão, os membros da o grupo 1 disputa o slot 1, etc. Fazendo uma divisão apropriada das estações em grupos, a quantidade de contenção para cada slot pode ser reduzida, operando cada slot próximo à extremidade esquerda da Fig. 4-9.

O truque é como atribuir estações a slots. Antes de olhar para o caso geral, vamos considerar alguns casos especiais. Em um extremo, cada grupo tem apenas um membro. Tal atribuição garante que nunca haverá colisões antes de porque no máximo uma estação está disputando um determinado slot. Nós vimos tal protocolos anteriores (por exemplo, contagem regressiva binária). O próximo caso especial é atribuir dois estações por grupo. A probabilidade de que ambos tentem transmitir durante um slot é p^2 , que para um p pequeno é desprezível. À medida que mais e mais estações são atribuídas a no mesmo slot, a probabilidade de uma colisão aumenta, mas o comprimento do mapa de bits varredura necessária para dar a todos uma chance de analistas. O caso limite é um único grupo contendo todas as estações (isto é, ALOHA com slot). O que precisamos é uma maneira de atribuir estações a slots dinamicamente, com muitas estações por slot quando a carga é baixa e poucas (ou mesmo apenas uma) estação por slot quando a carga é alta.

O Protocolo Adaptive Tree Walk

Uma maneira particularmente simples de realizar a tarefa necessária é usar o algoritmo desenvolvido pelo Exército dos EUA para testar soldados para sífilis durante Segunda Guerra Mundial (Dorfman, 1943). Em suma, o Exército coletou uma amostra de sangue de N soldados. Uma porção de cada amostra foi vertida em um único tubo de ensaio. Esta misturada amostra foi então testada para anticorpos. Se nenhum fosse encontrado, todos os soldados no grupo foram declarados saudáveis. Se os anticorpos estivessem presentes, duas novas amostras mistas

Página 300

276

O SUBCAMADA DE CONTROLE DE ACESSO MÉDIO INDIVÍDUO. 4

foram preparados, um dos soldados 1 a $N/2$ e um dos demais. O processo foi repetido recursivamente até que os soldados infectados fossem determinados. Para a versão computadorizada deste algoritmo (Capetanakis, 1979), é conveniente a pensar nas estações como as folhas de uma árvore binária, conforme ilustrado em Fig. 4-10. No primeiro slot de contenção após uma transmissão de quadro bem-sucedida, slot 0, todas as estações podem tentar adquirir o canal. Se um deles fizer muito bem. Se houver uma colisão, então, durante o slot 1, apenas as estações que se enquadram o nó 2 na árvore pode competir. Se um deles adquirir o canal, o slot segue a seguir, o quadro é reservado para as estações no nó 3. Se, por outro lado, duas ou mais estações no nó 2 desejam transmitir, haverá uma colisão durante o slot 1, caso em que é a vez do nó 4 durante o slot 2.

1
2
3
4
5
6
7
UMA
B
C
D
E
F
G
H

Estações

Figura 4-10. A árvore por oito estações.

Em essência, se ocorrer uma colisão durante o slot 0, toda a árvore é pesquisada, profundidade primeiro, para localizar todas as estações prontas. Cada slot de bit está associado a algum nó na árvore. Se ocorrer uma colisão, a pesquisa continua recursivamente com o filhos esquerdo e direito do nó. Se um slot de bit estiver ocioso ou se apenas uma estação transmitir nele, a busca de seu nó pode parar porque todas as estações prontas foram localizadas. (Se houvesse mais de um, teria ocorrido uma colisão.)

Quando a carga no sistema é pesada, dificilmente vale a pena o esforço de dedicar slot 0 ao nó 1 porque isso faz sentido apenas no caso improvável de que precisamente uma estação tem um quadro para enviar. Da mesma forma, pode-se argumentar que os nós 2 e 3 deve ser ignorado também pelo mesmo motivo. Em termos mais gerais, em que nível na árvore a pesquisa deve começar? Claramente, quanto mais pesada a carga, mais longe abaixo da árvore a busca deve começar. Vamos supor que cada estação tem uma boa estimativa do número de estações prontas, q , por exemplo, do monitoramento tráfego recente.

Para continuar, vamos numerar os níveis da árvore a partir do topo, com o nó 1 em Fig. 4-10 no nível 0, nós 2 e 3 no nível 1, etc. Observe que cada nó no nível i

Página 301

SEC. 4,2

PROTOCOLOS DE ACESSO MÚLTIPLOS

277

tem uma fração 2

- i das estações abaixo dele. Se as estações q prontas forem uniformemente distribuído, o número esperado deles abaixo de um nó específico no nível i é apenas 2

- $i q$. Intuitivamente, esperaríamos que o nível ideal para começar a pesquisar a árvore para ser aquele em que o número médio de estações concorrentes por slot é 1, ou seja, o nível em que 2

- $i q = 1$. Resolvendo essa equação, descobrimos que $i = \log_2 q$.

Numerosas melhorias no algoritmo básico foram descobertas e são discutido com algum detalhe por Bertsekas e Gallager (1992). Por exemplo, considere er o caso das estações G e H serem as únicas que desejam transmitir. No nó 1 ocorrerá uma colisão, então 2 serão testados e descobertos inativos. É inútil nó de sondagem 3, pois é garantido que haja uma colisão (sabemos que dois ou mais estações abaixo de 1 estão prontas e nenhuma delas está abaixo de 2, portanto, todas devem estar abaixo)

3). A sonda de 3 pode ser ignorada e 6 tentada em seguida. Quando esta sonda também aparece nada, 7 pode ser ignorado e o nó G tentado em seguida.

4.2.5 Protocolos de LAN sem fio

Um sistema de laptops que se comunicam por rádio pode ser considerado um LAN sem fio, como discutimos na Seç. 1.5.3. Tal LAN é um exemplo de um canal de transmissão. Ele também tem propriedades um pouco diferentes de uma LAN com fio, o que leva a diferentes protocolos MAC. Nesta seção, examinaremos alguns dos esses protocolos. Na seção 4.4, examinaremos 802.11 (WiFi) em detalhes.

Uma configuração comum para uma LAN sem fio é um prédio de escritórios com acesso pontos (APs) colocados estratégicamente ao redor do edifício. Os APs são conectados juntos usando cobre ou fibra e fornecer conectividade às estações que se comunicam com eles. E se a potência de transmissão dos APs e laptops é ajustada para ter um intervalo de dezenas de metros, os quartos próximos tornam-se como uma única célula e todo o edifício torna-se como os sistemas de telefonia celular que estudamos no cap. 2, exceto que cada célula tem apenas um canal. Este canal é compartilhado por todas as estações da célula, incluindo o AP. Ele normalmente fornece larguras de banda megabit / s, de até 600 Mbps.

Já observamos que os sistemas sem fio normalmente não podem detectar um colisão enquanto está ocorrendo. O sinal recebido em uma estação pode ser minúsculo, talvez um milhões de vezes mais fraco do que o sinal que está sendo transmitido. Encontrar é como

procurando uma ondulação no oceano. Em vez disso, os reconhecimentos são usados para descartar cobrir colisões e outros erros após o fato.

Há uma diferença ainda mais importante entre LANs sem fio e com fio LANs. Uma estação em uma LAN sem fio pode não ser capaz de transmitir quadros para ou receber frames de todas as outras estações por causa do alcance de rádio limitado da estação. Em LANs com fio, quando uma estação envia um quadro, todas as outras estações recebem isto. A ausência dessa propriedade em LANs sem fio causa uma variedade de complicações.

Faremos a suposição simplificada de que cada transmissor de rádio tem alguns alcance fixo, representado por uma região de cobertura circular dentro da qual outra estação não pode sentir e receber a transmissão da estação. É importante perceber que

Página 302

278

O SUBCAMADA DE CONTROLE DE ACESSO MÉDIO

INDIVÍDUO. 4

na prática, as regiões de cobertura não são tão regulares porque a propagação de os sinais de rádio dependem do ambiente. Paredes e outros obstáculos que atenuam absorver e refletir os sinais podem fazer com que o intervalo difira acentuadamente em diferentes direções

. Mas um modelo circular simples servirá para nossos propósitos.

Uma abordagem ingênua para usar uma LAN sem fio pode ser tentar CSMA: apenas ouça para outras transmissões e apenas transmitir se ninguém mais estiver fazendo isso. O problema é, este protocolo não é realmente uma boa maneira de pensar sobre wireless porque sinais de recepção são interferências no receptor, não no emissor. Para ver a natureza do problema, considere a Figura 4-11, onde quatro estações sem fio são ilustradas . Para nossos propósitos, não importa quais são APs e quais são laptops.

O alcance do rádio é tal que *A* e *B* estão dentro do alcance um do outro e podem potencializar interferem um no outro. *C* também pode interferir potencialmente com *B* e *D*, mas não com *um* .

Alcance de rádio

(uma)

(b)

Alcance de rádio

UMA

B

C

D

UMA

B

C

D

Figura 4-11. Uma LAN sem fio. (a) *A* e *C* são terminais ocultos quando transmitindo para *B*. (b) *B* e *C* estão expostos terminais durante a transmissão para *um* e *D* .

Considere primeiro o que acontece quando *A* e *C* transmitem para *B* , conforme representado em Fig. 4-11 (a). Se *A* enviar e *C* imediatamente sentir o meio, ele não ouvir *A* porque *A* está fora do alcance. Assim, *C* vai concluir falsamente que pode transmitir para *B* . Se *C* começar a transmitir, ele interferirá em *B* , apagando o quadro de *Um* . (Assumimos aqui que nenhum esquema do tipo CDMA é usado para fornecer vários canais, então as colisões distorcem o sinal e destroem os dois quadros.) Queremos um Protocolo MAC que impedirá que este tipo de colisão aconteça porque desperdiça largura de banda. O problema de uma estação não ser capaz de detectar um potencial competidor para o meio porque o competidor está muito longe é chamado de **problema de terminal oculto** .

Agora, vejamos uma situação diferente: *B* transmitindo para *A* ao mesmo tempo que *C* deseja transmitir para *D* , como mostra a Figura 4.11 (b). Se *C* sentir o meio, ouvirá uma transmissão e concluirá falsamente que pode não ser enviada para *D* (mostrado como uma linha tracejada). Na verdade, tal transmissão causaria má recepção apenas na zona entre *B* e *C* , onde nenhum dos receptores pretendidos está localizado. Nós quer um protocolo MAC que evita que esse tipo de adiamento aconteça porque ele desperdiça largura de banda. O problema é denominado **problema de terminal exposto** .

A dificuldade é que, antes de iniciar uma transmissão, a emissora quer muito saber se há atividade de rádio ao redor do receptor. CSMA apenas diz isso

Página 303

SEC. 4,2

PROTÓCOLOS DE ACESSO MÚLTIPLOS

279

se há atividade perto do transmissor detectando a portadora. Com um fio, todos os sinais se propagam para todas as estações, portanto, essa distinção não existe. Contudo, apenas uma transmissão pode ocorrer de uma vez em qualquer lugar do sistema. Em um sistema baseado em ondas de rádio de curto alcance, várias transmissões podem ocorrer simultaneamente

simultaneamente se todos eles tiverem destinos diferentes e esses destinos estiverem fora de alcance um do outro. Queremos que essa simultaneidade aconteça conforme a célula fica maior e maiores, da mesma forma que as pessoas em uma festa não devem esperar por todos em a sala fique em silêncio antes de eles falarem; várias conversas podem ocorrer em uma vez em uma sala grande, desde que não sejam direcionados para o mesmo local.

Um protocolo inicial e influente que aborda esses problemas para redes sem fio LANs é **MACA** (**M**ultiple **A**ccess with **C**ollision **A**voidance) (Karn, 1990). A ideia básica por trás disso é que o emissor estimule o receptor a emitir um quadro curto, para que as estações próximas possam detectar esta transmissão e evitar a transmissão durante o próximo quadro de dados (grande). Esta técnica é usada ao invés de sentido de portador.

MACA é ilustrado na Figura 4-12. Vejamos como *A* envia um quadro para *B*. *UMA* começa enviando um quadro **RTS** (**R**equest **T**o **S**end) para *B*, como mostra a Figura 4.12 (a). Este quadro curto (30 bytes) contém o comprimento do quadro de dados que irá eventualmente almente siga. Então *B* responde com um quadro **CTS** (**C**lear **T**o **S**end), conforme mostrado em Fig. 4-12 (b). O quadro CTS contém o comprimento dos dados (copiados do RTS quadro, Armação). Após o recebimento do quadro CTS, *A* começa a transmissão.

(uma)

(b)

Alcance do transmissor de A

A RTS

E

B

D

C

UMA

CTS

E

B

D

C

Alcance do transmissor de B

Figura 4-12. O protocolo MACA. (a) *Um* enviar um RTS para *B*. (b) *B* respondendo com um CTS para *um*.

Agora vamos ver como as estações que ouvem esses quadros reagem. Qualquer estação ouvindo o RTS está claramente perto de *A* e deve permanecer em silêncio por tempo suficiente para que o CTS seja transmitido de volta para *A* sem conflito. Qualquer estação ouvindo o CTS está claramente perto de *B* e deve permanecer em silêncio durante a próxima transferência de dados missão, cujo comprimento pode determinar examinando o quadro CTS .

Página 304

280

O SUBCAMADA DE CONTROLE DE ACESSO MÉDIO

INDIVÍDUO. 4

Na Fig. 4-12, *C* está dentro do alcance de *um*, mas não dentro do alcance de *B*. Portanto, ouve o RTS de *A*, mas não a CTS de *B*. Contanto que não interfira com o CTS, está livre para transmitir enquanto o quadro de dados está sendo enviado. Em contraste, *D* é dentro do alcance de *B*, mas não *um*. Ele não ouve o RTS, mas ouve o CTS .

Ouvir o CTS avisa que está perto de uma estação que está prestes a receber um quadro, então adia o envio de qualquer coisa até que seja esperado que o quadro seja concluído. A estação *E* ouve ambas as mensagens de controle e, como *D*, deve ficar em silêncio até que os dados o quadro está completo.

Apesar dessas precauções, ainda podem ocorrer colisões. Por exemplo, *B* e *C* poderia enviar quadros RTS para *A* ao mesmo tempo. Eles colidirão e serão perdidos. Em caso de colisão, um transmissor malsucedido (ou seja, aquele que não ouve um CTS dentro do intervalo de tempo esperado) espera uma quantidade aleatória de tempo e tenta de novo mais tarde.

4.3 ETHERNET

Agora terminamos nossa discussão sobre protocolos de alocação de canais no abstrato, então é hora de ver como esses princípios se aplicam a sistemas reais. Muitos os projetos de redes pessoais, locais e de área metropolitana têm sido padronizados dardizado sob o nome de IEEE 802. Alguns sobreviveram, mas muitos não, como vimos na Figura 1-38. Algumas pessoas que acreditam em reencarnação pensam que Charles Darwin voltou como membro da IEEE Standards Association para eliminar os inaptos. Os mais importantes dos sobreviventes são 802.3 (Ethernet) e 802.11 (LAN sem fio). Bluetooth (PAN sem fio) é amplamente implantado, mas agora foi padronizado fora do 802.15. Com 802.16 (MAN sem fio), é muito cedo contar. Consulte a 6^a edição deste livro para descobrir.

Começaremos nosso estudo de sistemas reais com Ethernet, provavelmente o mais ubiquitous tipo de rede de computadores do mundo. Existem dois tipos de Ethernet: **classic Ethernet**, que resolve o problema de acesso múltiplo usando as técnicas que estudou neste capítulo; e **Ethernet comutada**, em que dispositivos chamados switches são usados para conectar computadores diferentes. É importante notar que, embora ambos sejam chamados de Ethernet, são bastante diferentes. Classic Ethernet é a forma original e funcionou a taxas de 3 a 10 Mbps. Ethernet comutada é o que a Ethernet se tornou e funciona a 100, 1000 e 10.000 Mbps, em formulários de chamada ed fast Ethernet, gigabit Ethernet e 10 gigabit Ethernet. Na prática, apenas Ethernet comutada é usada atualmente.

Discutiremos essas formas históricas de Ethernet em ordem cronológica mostrando como eles se desenvolveram. Uma vez que Ethernet e IEEE 802.3 são idênticos, exceto por uma pequena diferença (que discutiremos em breve), muitas pessoas usam os termos "Ethernet" e "IEEE 802.3" de forma intercambiável. Faremos isso também. Para mais informações sobre Ethernet, consulte Spurgeon (2000).

4.3.1 Camada Física Ethernet Clássica

A história da Ethernet começa quase ao mesmo tempo que a da ALOHA, quando um estudante chamado Bob Metcalfe obteve seu diploma de bacharel no MIT e depois mudou-se rio acima para obter seu doutorado. em Harvard. Durante seus estudos, ele foi exposto a Trabalho de Abramson. Ele ficou tão interessado nisso que depois de se formar na Harvard, ele decidiu passar o verão no Havaí trabalhando com Abramson antes começando a trabalhar na Xerox PARC (Palo Alto Research Center). Quando ele conseguiu PARC, ele viu que os pesquisadores de lá haviam projetado e construído o que mais tarde ser chamados de computadores pessoais. Mas as máquinas estavam isoladas. Usando seu conhecimento

ponta do trabalho de Abramson, ele, junto com seu colega David Boggs, projetou e implementou a primeira rede local (Metcalfe e Boggs, 1976). Costumava um único cabo coaxial longo e grosso e funcionava a 3 Mbps.

Eles chamado o sistema **Ethernet** após o *éter luminoso*, através do qual Já se pensou que a radiação eletromagnética se propagava. (Quando o século 19 O físico britânico James Clerk Maxwell descobriu que a radiação eletromagnética

poderia ser descrito por uma equação de onda, os cientistas presumiram que o espaço deve ser preenchido com algum meio etéreo no qual a radiação estava se propagando. Somente após o famoso experimento Michelson-Morley em 1887, os físicos descobriram que a radiação eletromagnética pode se propagar no vácuo.)

A Ethernet da Xerox foi tão bem-sucedida que a DEC, a Intel e a Xerox elaboraram um padrão em 1978 para uma Ethernet de 10 Mbps, chamado de **padrão DIX**. Com menor mudança, o padrão DIX tornou-se o padrão IEEE 802.3 em 1983. Unfornecidamente para a Xerox, ela já tinha um histórico de fazer invenções seminais (como o computador pessoal) e, em seguida, deixar de comercializá-los, uma história contada em *Fumbling the Future* (Smith e Alexander, 1988). Quando a Xerox mostrou pouco interesse em fazer qualquer coisa com Ethernet além de ajudar a padronizá-la, Metcalfe formou sua própria empresa, a 3Com, para vender adaptadores Ethernet para PCs. isto vendeu muitos milhões deles.

Ethernet clássica serpenteava pelo prédio como um único cabo longo para o qual todos os computadores foram conectados. Essa arquitetura é mostrada na Figura 4-13. O primeiro variedade, popularmente chamada de **Ethernet espessa**, parecia uma mangueira de jardim amarela, com marcações a cada 2,5 metros para mostrar onde conectar os computadores. (O padrão 802.3 não exigia que o cabo fosse amarelo, mas sugeriu isso.)

conseguido por **Ethernet fina**, que se dobrou mais facilmente e fez conexões usando conectores BNC padrão da indústria. Thin Ethernet era muito mais barato e fácil para instalar, mas poderia funcionar por apenas 185 metros por segmento (em vez de 500 m com Ethernet espessa), cada uma das quais suportava apenas 30 máquinas (em vez de 100). Cada versão de Ethernet tem um comprimento máximo de cabo por segmento (ou seja, comprimento não amplificado) ao longo do qual o sinal se propagará. Para permitir uma rede maior funciona, vários cabos podem ser conectados por **repetidores**. Um repetidor é um físico dispositivo de camada que recebe, amplifica (ou seja, regenera) e retransmite sinais em ambas direções. No que diz respeito ao software, uma série de segmentos de cabo

Página 306

282

O SUBCAMADA DE CONTROLE DE ACESSO MÉDIO INDIVÍDUO. 4

Éter
Transceptor
Interface
cabos

Figura 4-13. Arquitetura da Ethernet clássica.

conectado por repetidores não é diferente de um único cabo (exceto por um pequeno quantidade de atraso introduzida pelos repetidores).

Em cada um desses cabos, as informações foram enviadas usando a codificação Manchester que estudamos na Seç. 2.5. Uma Ethernet pode conter vários segmentos de cabo e vários repetidores, mas nenhum transceptor poderia estar a mais de 2,5 km um do outro e nenhum caminho entre dois transceptores poderia atravessar mais de quatro repetidores.

A razão para essa restrição era que o protocolo MAC, que veremos a seguir, funcionaria corretamente.

4.3.2 Protocolo de subcamada MAC Ethernet clássico

O formato usado para enviar quadros é mostrado na Figura 4-14. Primeiro vem um *pré - treinamento ble* de 8 bytes, cada uma contendo o padrão de bits 10101010 (com a exceção da último byte, em que os últimos 2 bits são definidos como 11). Este último byte é chamado de *íncio de*

Delimitador de *quadro* para 802.3. A codificação Manchester deste padrão produz um Onda quadrada de 10 MHz para 6,4 µs para permitir que o relógio do receptor sincronize com o remetente. Os últimos dois bits 1 dizem ao receptor que o resto do quadro é

Prestes a começar.

Preâmbulo
(uma)
Bytes
Tipo
Dados

Almofada	
Verifica-	
soma	
Destino	
endereço	
Fonte	
endereço	
8	
2	
0-1500	
0-46	
4	
6	
6	
Preâmbulo	
(b)	
comprimento	
Dados	
Almofada	
Verifica-	
soma	
Destino	
endereço	
Fonte	
endereço	

Figura 4-14. Formatos de quadro. (a) Ethernet (DIX). (b) IEEE 802.3.

Em seguida, vêm dois endereços, um para o destino e outro para a origem. Eles cada um tem 6 bytes de comprimento. O primeiro bit transmitido do endereço de destino é 0 para

Página 307

SEC. 4,3
ETHERNET

283

endereços comuns e 1 para endereços de grupo. Endereços de grupo permitem vários estações para ouvir um único endereço. Quando um quadro é enviado para um endereço de grupo, todos

as estações do grupo o recebem. O envio para um grupo de estações é denominado **multifunção**. O endereço especial que consiste em todos os bits 1 é reservado para **transmissão**.

Um quadro contendo todos os 1s no campo de destino é aceito por todas as estações na rede. O multicast é mais seletivo, mas envolve gerenciamento de grupo para definir quais estações estão no grupo. Por outro lado, a transmissão não difere

diferenciar entre as estações em tudo, portanto, não requer nenhum gerenciamento de grupo.

Uma característica interessante dos endereços de origem da estação é que eles são globalmente único, atribuído centralmente pelo IEEE para garantir que não haja duas estações em qualquer lugar do

mundo tem o mesmo endereço. A ideia é que qualquer estação pode atender exclusivamente qualquer outra estação apenas fornecendo o número correto de 48 bits. Para fazer isso, os 3 primeiros

bytes do campo de endereço são usados para um **OUI** (**Organizationally Unique Identifier**). Os valores desse campo são atribuídos pelo IEEE e indicam um fabricante.

Os fabricantes são atribuídos a blocos de 2²⁴ endereços. O fabricante atribui os últimos 3 bytes do endereço e programa o endereço completo na NIC antes é vendido.

Em seguida, vem o campo *Tipo* ou *Comprimento*, dependendo se o quadro é Ethernet ou IEEE 802.3. Ethernet usa um campo *Tipo* para dizer ao receptor o que fazer com o quadro. Vários protocolos de camada de rede podem estar em uso ao mesmo tempo na mesma máquina, então quando um frame Ethernet chega, o sistema operacional tem para saber a qual entregar a moldura. O campo *Tipo* especifica qual processo para dar o quadro. Por exemplo, um código de tipo de 0x0800 significa que os dados contém um pacote IPv4.

IEEE 802.3, em sua sabedoria, decidiu que este campo carregaria o comprimento de o quadro, uma vez que o comprimento da Ethernet foi determinado olhando dentro dos dados - um violação de camadas, se é que alguma vez houve. Claro, isso significava que não havia como para o receptor descobrir o que fazer com um quadro de entrada. Esse problema foi tratado pela adição de outro cabeçalho para o **LLC** (**Logical Link Con-**

trol) protocolo nos dados. Ele usa 8 bytes para transmitir os 2 bytes do protocolo digitie informações.

Infelizmente, na época em que 802.3 foi publicado, tanto hardware e software para DIX Ethernet já estava em uso por poucos fabricantes e usuários estavam entusiasmados em reempacotar os campos *Tipo* e *Comprimento*. Em 1997, IEEE jogou a toalha e disse que as duas coisas estavam bem com ele. Felizmente, todos os Os campos de *tipo* em uso antes de 1997 tinham valores maiores que 1500, então bem estabelecidos como o tamanho máximo dos dados. Agora, a regra é que qualquer número menor que ou igual a 0x600 (1536) pode ser interpretado como *Comprimento* e qualquer número maior que 0x600 pode ser interpretado como *tipo*. Agora o IEEE pode afirmar que todos estão usando seu padrão e todos os outros podem continuar fazendo o que já estavam fazendo (não se preocupando com LLC) sem se sentir culpado por isso.

Em seguida, vêm os dados, com até 1.500 bytes. Este limite foi escolhido um tanto arbitrariamente, na época, o padrão Ethernet foi lançado em pedra, principalmente com base no fato

Página 308

284

O SUBCAMADA DE CONTROLE DE ACESSO MÉDIO INDIVÍDUO. 4

que um transceptor precisa de RAM suficiente para armazenar um quadro inteiro e a RAM foi cara em 1978. Um limite superior maior significaria mais RAM e, portanto, um transceptor mais caro.

Além de haver um comprimento máximo de quadro, também há um mínimo comprimento do quadro. Embora um campo de dados de 0 bytes às vezes seja útil, ele causa um problema

lem. Quando um transceptor detecta uma colisão, ele trunca o quadro atual, que significa que pedaços perdidos de frames aparecem no cabo o tempo todo. Para tornar mais fácil distinguir quadros válidos de lixo, a Ethernet requer que quadros válidos devem ter pelo menos 64 bytes de comprimento, do endereço de destino até a soma de verificação,

incluindo ambos. Se a porção de dados de um quadro for menor que 46 bytes, o campo *Pad* é usado para preencher o quadro até o tamanho mínimo.

Outra razão (e mais importante) para ter um quadro de comprimento mínimo é evitar que uma estação complete a transmissão de um pequeno quadro antes do o primeiro bit chegou até a extremidade do cabo, onde pode colidir com outro quadro. Esse problema é ilustrado na Figura 4-15. No tempo 0, estação *A*, em uma extremidade da rede, envia um quadro. Vamos chamar o tempo de propagação para este quadro para alcançar a outra extremidade τ . Pouco antes de o quadro chegar ao outro lado (isto é, no tempo $\tau - \epsilon$), a estação mais distante, *B*, começa a transmitir. Quando *B* detecta que está recebendo mais energia do que emitindo, sabe que uma colisão ocorreu, então ele aborta sua transmissão e gera uma explosão de ruído de 48 bits para avisar todas as outras estações. Em outras palavras, ele congestioniza o éter para garantir que o remetente faça

não perca a colisão. Por volta do tempo 2τ , o remetente vê a explosão de ruído e aborta sua transmissão também. Em seguida, ele espera um tempo aleatório antes de tentar novamente.

O pacote começa no tempo 0

UMA

B

UMA

B

Pacote quase

no morcego

Colisão em

Tempo

UMA

B

Explosão de ruído fica

de volta para A em 2

UMA

B

(uma)

(b)

(c)

(d)

Figura 4-15. A detecção de colisão pode demorar até 2τ .

Se uma estação tenta transmitir um quadro muito curto, é concebível que um colisão ocorrerá, mas a transmissão terá sido concluída antes do estouro de ruído volta para a estação às 2τ . O remetente, então, concluirá incorretamente que o quadro foi enviado com sucesso. Para evitar que essa situação ocorra, todos os quadros deve demorar mais do que 2τ para enviar de modo que a transmissão ainda esteja ocorrendo quando

SEC. 4,3

ETHERNET

285

a explosão de ruído volta para o remetente. Para uma LAN de 10 Mbps com um máximo comprimento de 2500 metros e quatro repetidores (da especificação 802.3), o tempo de ida e volta (incluindo o tempo de propagação pelos quatro repetidores) foi determinado como sendo quase 50 μ seg no pior caso. Portanto, o menor permitido o quadro deve levar pelo menos esse tempo para ser transmitido. A 10 Mbps, um bit leva 100 nseg, então

500 bits é o menor quadro com garantia de funcionamento. Para adicionar alguma margem de segurança, esse número foi arredondado para 512 bits ou 64 bytes.

O campo final é o *Checksum*. É um CRC de 32 bits do tipo que estudamos em Sec. 3.2. Na verdade, é definido exatamente pelo polinômio gerador que demos lá, que apareceu para PPP, ADSL e outros links também. Este CRC é um erro-deteção de código que é usado para determinar se os bits do quadro foram recebidos corretamente. Ele apenas faz a detecção de erros, com o quadro descartado se um erro for detectado.

CSMA / CD com Binary Exponential Backoff

A Ethernet clássica usa o algoritmo CSMA / CD 1 persistente que estudamos em Sec. 4.2. Este descriptor significa apenas que as estações detectam o meio quando tenha um quadro para enviar e envie o quadro assim que a mídia ficar ociosa.

Eles monitoram o canal em busca de colisões enquanto enviam. Se houver uma colisão, eles abortar a transmissão com um curto sinal de congestionamento e retransmitir após um intervalo.

Vamos agora ver como o intervalo aleatório é determinado quando uma colisão ocorre, pois é um novo método. O modelo ainda é o da Figura 4-5. Depois de uma colisão, o tempo é dividido em slots discretos, cujo comprimento é igual à rodada de pior caso tempo de propagação da viagem no éter (2τ). Para acomodar o caminho mais longo permitido pela Ethernet, o tempo de slot foi definido para 512 tempos de bits ou 51,2 μ seg.

Após a primeira colisão, cada estação espera 0 ou 1 slot times aleatoriamente antes de tentar novamente. Se duas estações colidem e cada uma escolhe a mesma aleatoriamente número, eles vão colidir novamente. Após a segunda colisão, cada um escolhe 0, 1, 2 ou 3 aleatoriamente e espera esse número de vezes de slot. Se uma terceira colisão ocorre (a probabilidade de isso acontecer é de 0,25), da próxima vez que o número de slots a aguardar são escolhidos aleatoriamente no intervalo 0 a $2^3 - 1$.

Em geral, após i colisões, um número aleatório entre 0 e 2^{i-1} é escolhido, e esse número de slots é ignorado. No entanto, após 10 colisões foram alcançado, o intervalo de randomização é congelado em um máximo de 1023 slots. Depois de 16 colisões, o controlador joga a toalha e relata a falha de volta para o computador. A recuperação posterior depende das camadas superiores.

Este algoritmo, chamado **backoff exponencial binário**, foi escolhido para adaptar-se adequadamente ao número de estações que tentam enviar. Se o intervalo de randomização fosse todas as colisões foram 1023, a chance de duas estações colidirem pela segunda vez seria insignificante, mas a espera média após uma colisão seria de centenas de tempos de slot, introduzindo um atraso significativo. Por outro lado, se cada estação sempre

286

O SUBCAMADA DE CONTROLE DE ACESSO MÉDIO INDIVÍDUO. 4

atrasado para 0 ou 1 slots, então se 100 estações já tentaram enviar ao mesmo tempo, iria colidir continuamente até que 99 deles pegassem 1 e a estação restante escolheu 0. Isso pode levar anos. Fazendo o intervalo de randomização crescer exponencialmente, à medida que mais e mais colisões consecutivas ocorrem, o algoritmo garante um atraso baixo quando apenas algumas estações colidem, mas também garante que as colisões são resolvidos em um intervalo razoável quando muitas estações colidem. Truncando o backoff em 1023 impede que o limite fique muito grande.

Se não houver colisão, o remetente presume que o quadro provavelmente foi bem-sucedido entregue com sucesso. Ou seja, nem CSMA / CD nem Ethernet fornecem reconhecimento de erros. Esta escolha é apropriada para canais de fibra óptica e com fio que têm baixas taxas de erro. Quaisquer erros que ocorram devem ser detectados pelo CRC e recuperado por camadas superiores. Para canais sem fio com mais erros, nós verá que são usados reconhecimentos.

4.3.3 Desempenho Ethernet

Agora vamos examinar brevemente o desempenho da Ethernet clássica sob condições de carga pesada e constante, ou seja, com k estações sempre prontas para transmitir. Uma análise rigorosa do algoritmo de backoff exponencial binário é complicada. Em vez disso, seguiremos Metcalfe e Boggs (1976) e assumiremos uma constante probabilidade de retransmissão em cada slot. Se cada estação transmitir durante um contention slot de conexão com probabilidade p , a probabilidade A de que alguma estação adquire a mudança nel nesse slot é

$$A = kp(1 - p)^{k-1} \quad (4-5)$$

A é maximizado quando $p = 1/k$, com $A \rightarrow 1/e$ como $k \rightarrow \infty$. A probabilidade de que intervalo de contenção tem exatamente j slots em que é $A(1 - A)^{j-1}$, então o número médio de slots por contenção é dado por

$$\sum_{j=0}^{\infty} jA(1 - A)^{j-1} = UMA$$

Como cada slot tem uma duração 2τ , o intervalo de contenção médio, w , é $2\tau/A$. Assumindo p ótimo, o número médio de slots de contenção nunca é mais do que e , então w é no máximo $2\tau e \sim 5,4\tau$.

Se o quadro médio leva P seg para transmitir, quando muitas estações têm quadros para enviar,

Eficiência do canal =

$$P + 2\tau/A$$
$$P$$
$$(4-6)$$

Aqui vemos onde entra a distância máxima do cabo entre quaisquer duas estações nas figuras de desempenho. Quanto mais longo o cabo, mais longa será a contenção intervalo, é por isso que o padrão Ethernet especifica um comprimento máximo de cabo.

287

É instrutivo formular a Eq. (4-6) em termos de comprimento de quadro, F , a rede largura de banda de trabalho, B , o comprimento do cabo, L , e a velocidade de propagação do sinal, c , para o caso ideal de slots de contenção e por quadro. Com $P = F/B$, Eq. (4-6)

torna-se

$$\text{Eficiência do canal} = \frac{1 + 2 BLe / cF}{1}$$

(4-7)

Quando o segundo termo do denominador for grande, a eficiência da rede será baixa.

Mais especificamente, aumentando a largura de banda ou distância da rede (o produto BL) reduz a eficiência para um determinado tamanho de quadro. Infelizmente, muita pesquisa na rede hardware de trabalho visa justamente aumentar este produto. As pessoas querem alto largura de banda em longas distâncias (MANs de fibra óptica, por exemplo), ainda éter clássico. A rede implementada dessa maneira não é o melhor sistema para esses aplicativos. Nós verá outras maneiras de implementar Ethernet na próxima seção.

Na Fig. 4-16, a eficiência do canal é plotada em relação ao número de postos prontos para $2\tau = 51,2 \mu\text{seg}$ e uma taxa de dados de 10 Mbps, usando a Eq. (4-7). Com um 64-byte slot time, não é surpreendente que os quadros de 64 bytes não sejam eficientes. No Por outro lado, com quadros de 1024 bytes e um valor assintótico de e ranhuras de 64 bytes por intervalo de contenção, o período de contenção tem 174 bytes de comprimento e a eficiência é 85%. Este resultado é muito melhor do que a eficiência de 37% do ALOHA com fenda.

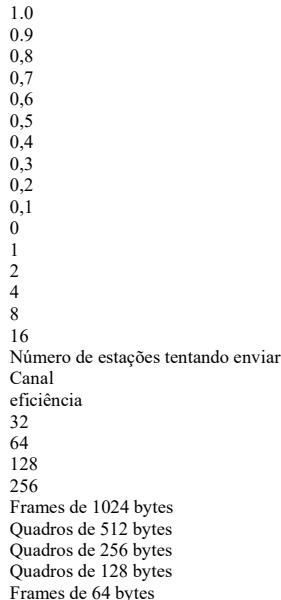


Figura 4-16. Eficiência da Ethernet a 10 Mbps com slots de 512 bits.

Provavelmente, vale a pena mencionar que tem havido uma grande quantidade de teorias análise de desempenho cal de Ethernet (e outras redes). A maioria dos resultados deve ser tomado com um grão (ou melhor ainda, uma tonelada) de sal, por dois motivos.

Página 312

288

O SUBCAMADA DE CONTROLE DE ACESSO MÉDIO INDIVÍDUO. 4

Em primeiro lugar, praticamente todo o trabalho teórico assume o tráfego de Poisson. Como pesquisadores começaram a olhar para dados reais, agora parece que o tráfego de rede raramente é Poisson. Em vez disso, é auto-semelhante ou intermitente ao longo de uma gama de escalas de tempo (Paxson e Floyd, 1995; e Leland et al., 1994). O que isso significa é que calcular a média longos períodos de tempo não suavizam o tráfego. Além de usar perguntas modelos capazes, muitas das análises enfocam os casos de desempenho "interessantes" de carga anormalmente alta. Boggs et al. (1988) mostrou por experimentação que Ethernet funciona bem na realidade, mesmo com carga moderadamente alta.

4.3.4 Ethernet Comutada

Ethernet logo começou a evoluir para longe da arquitetura de cabo longo único de Ethernet clássica. Os problemas associados à localização de rupturas ou conexões soltas ções o levaram a um tipo diferente de padrão de fiação, em que cada estação tem um cabo dedicado passando para um **hub** central . Um hub simplesmente conecta todos os fios eletricamente, como se estivessem soldados juntos. Esta configuração é mostrada na Fig. 4-17 (a).

Porta
Linha
Cubo
Interruptor
(uma)
(b)
Porta
Linha

Figura 4-17. (um hub. (b) Mudar.

Os fios eram pares trançados da companhia telefônica, já que a maioria dos prédios de escritórios já estavam ligados desta forma e normalmente havia muitas peças sobressalentes disponíveis. Isto reutilizar foi uma vitória, mas reduziu o comprimento máximo de cabo do hub para 100 metros (200 metros se pares trançados de categoria 5 de alta qualidade foram usados). Adicionando ou

remover uma estação é mais simples nesta configuração, e quebras de cabo podem ser detectado facilmente. Com as vantagens de poder usar a fiação existente e facilidade de manutenção, hubs de par trançado rapidamente se tornaram a forma dominante de Ethernet.

No entanto, os hubs não aumentam a capacidade porque são logicamente equivalentes ao único cabo longo da Ethernet clássica. À medida que mais e mais estações são adicionadas, cada estação obtém uma parcela decrescente da capacidade fixa. Eventualmente, a LAN vai saturar. Uma saída é ir para uma velocidade mais alta, digamos, de 10 Mbps para 100 Mbps, 1 Gbps ou velocidades ainda mais altas. Mas com o crescimento da multimídia e servidores poderosos, até mesmo uma Ethernet de 1 Gbps pode ficar saturada.

Página 313

SEC. 4,3
ETHERNET
289

Felizmente, existe uma outra maneira de lidar com o aumento da carga: comutada Ethernet. O coração deste sistema é um **switch** contendo um backplane de alta velocidade que conecta todas as portas, conforme mostrado na Figura 4-17 (b). Do lado de fora, um interruptor parece um hub. Ambos são caixas, normalmente com 4 a 48 portas, cada uma com um conector RJ-45 padrão para um cabo de par trançado. Cada cabo conecta o switch ou hub para um único computador, conforme mostrado na Fig. 4-18. Um switch tem o mesmo

vantagens como hub também. É fácil adicionar ou remover uma nova estação conectando ou desconectar um fio, e é fácil encontrar a maioria das falhas, uma vez que um cabo ou porta escamosa geralmente afetam apenas uma estação. Ainda há um componente compartilhado que pode falhar - o mudar-se, mas se todas as estações perderem a conectividade, o pessoal de TI saberá o que fazer para

conserte o problema: substitua todo o switch.

Interruptor
Par trançado
Alternar portas
Cubo

Figura 4-18. Um switch Ethernet.

Dentro da troca, entretanto, algo muito diferente está acontecendo. Comuta somente os quadros de saída para as portas para as quais esses quadros são destinados. Quando um a porta do switch recebe um quadro Ethernet de uma estação, o switch verifica o Ether-endereços de rede para ver a qual porta o quadro se destina. Esta etapa requer o switch para poder descobrir quais portas correspondem a quais endereços, um processo que descreveremos na Seç. 4.8 quando chegarmos ao caso geral de interruptores conectado a outros interruptores. Por enquanto, apenas suponha que o switch conhece o porta de destino do quadro. O switch então encaminha o quadro ao longo de sua alta velocidade backplane para a porta de destino. O backplane normalmente funciona a muitos Gbps,

usando um protocolo proprietário que não precisa ser padronizado porque é totalmente escondido dentro do switch. A porta de destino então transmite o quadro o fio para que alcance a estação pretendida. Nenhuma das outras portas mesmo sabe que a moldura existe.

O que acontece se mais de uma das estações ou portas quiser enviar um quadro ao mesmo tempo? Novamente, os switches são diferentes dos hubs. Em um hub, todas as estações estão em o mesmo **domínio de colisão**. Eles devem usar o algoritmo CSMA / CD para agendar suas transmissões. Em um switch, cada porta é sua própria colisão independente domínio. No caso comum em que o cabo é full duplex, tanto a estação quanto o porta pode enviar um quadro no cabo ao mesmo tempo, sem se preocupar com outros portos e estações. As colisões agora são impossíveis e CSMA / CD não é necessário. No entanto, se o cabo for half duplex, a estação e a porta devem disputar transmissão com CSMA / CD da maneira usual.

Página 314

290

O SUBCAMADA DE CONTROLE DE ACESSO MÉDIO INDIVÍDUO. 4

Um switch melhora o desempenho de um hub de duas maneiras. Primeiro, uma vez que existem sem colisões, a capacidade é usada com mais eficiência. Em segundo lugar, e mais importante entretanto, com um switch, vários quadros podem ser enviados simultaneamente (por diferentes estações).

Esses quadros alcançarão as portas do switch e passarão pela parte traseira do switch plano a ser produzido nas portas adequadas. No entanto, uma vez que dois quadros podem ser enviados

para a mesma porta de saída ao mesmo tempo, o switch deve ter buffer para que pode enfileirar temporariamente um quadro de entrada até que ele possa ser transmitido para a porta de saída.

No geral, essas melhorias fornecem uma grande vitória de desempenho que não é possível com um hub. A taxa de transferência total do sistema pode muitas vezes ser aumentada em uma ordem de

magnitude, dependendo do número de portas e padrões de tráfego.

A mudança nas portas de saída dos quadros também traz benefícios de segurança.

A maioria das interfaces LAN tem um **modo promíscuo**, no qual *todos os* quadros são dados a cada computador, não apenas aqueles endereçados a ele. Com um hub, cada computador que está anexado pode ver o tráfego enviado entre todos os outros computadores. Espiões e intrometidos adoram esse recurso. Com um switch, o tráfego é encaminhado apenas para as portas onde está destinado. Esta restrição fornece melhor isolamento para que o tráfego não é fácil escapar e cair nas mãos erradas. No entanto, é melhor criptografar tráfego se a segurança for realmente necessária.

Como o switch espera apenas frames Ethernet padrão em cada porta de entrada, é possível usar algumas das portas como concentradores. Na Fig. 4-18, a porta em o canto superior direito não está conectado a uma única estação, mas a um hub de 12 portas em vez de. Conforme os quadros chegam ao hub, eles lutam pelo éter da maneira usual, incluindo colisões e backoff binário. Quadros bem-sucedidos passam pelo hub para o switch e são tratados como quaisquer outros frames de entrada. O switch não sabe que eles tiveram que lutar para entrar. Uma vez no switch, eles são enviado para a linha de saída correta no painel traseiro de alta velocidade. Também é possível se o destino correto era um das linhas conectadas ao hub, caso em que o quadro já foi entregue, então o switch simplesmente o descarta. Hubs são mais simples e mais baratos do que os interruptores, mas devido à queda nos preços dos interruptores, eles se tornaram um espécies em perigo. As redes modernas usam amplamente Ethernet comutada. Nunca menos, hubs legados ainda existem.

4.3.5 Fast Ethernet

Ao mesmo tempo em que os switches estavam se tornando populares, a velocidade de 10 Mbps

Ethernet estava sob pressão. No início, 10 Mbps parecia o paraíso, apenas já que os modems a cabo pareciam o paraíso para os usuários de modems telefônicos. Mas o a novidade passou rapidamente. Como uma espécie de corolário da Lei de Parkinson (" Trabalho expande-se para preencher o tempo disponível para a sua conclusão "), parecia que os dados expandido para preencher a largura de banda disponível para sua transmissão. Muitas instalações precisavam de mais largura de banda e, portanto, tinham vários 10 Mbps LANs conectadas por um labirinto de repetidores, hubs e switches, embora à rede gerentes de trabalho às vezes sentiam que estavam sendo mantidos juntos por uma bolha

Página 315

SEC. 4,3
ETHERNET

291

goma e tela de frango. Mas mesmo com switches Ethernet, a largura de banda máxima de um único computador era limitado pelo cabo que o conectava à porta do switch. Foi neste ambiente que o IEEE reuniu o comitê 802.3 em 1992 com instruções para criar uma LAN mais rápida. Uma proposta era manter 802.3 exatamente como estava, mas apenas faça com que vá mais rápido. Outra proposta era refazer tudo - e dar a ele muitos recursos novos, como tráfego em tempo real e voz digitalizada, mas apenas mantenha o nome antigo (por razões de marketing). Depois de algumas discussões, o comitê decidiu manter o 802.3 do jeito que estava e apenas torná-lo mais rápido. Este estratégia faria o trabalho antes que a tecnologia mudasse e evitaria imprevistos problemas com um novo design. O novo design também seria retrógrado compatível com LANs Ethernet existentes. As pessoas por trás da proposta perdida fez o que qualquer pessoa da indústria de computadores que se preze teria feito sob estas circunstâncias: eles pisaram fora e formaram seu próprio comitê e standard de qualquer maneira sua LAN (eventualmente como 802.12). Ele fracassou miseravelmente. O trabalho foi feito rapidamente (pelas normas dos comitês de padrões), e o resultado, 802.3u, foi aprovado pelo IEEE em junho de 1995. Tecnicamente, 802.3u não é um novo padrão, mas um adendo ao padrão 802.3 existente (para enfatizar sua parte posterior compatibilidade de ala). Essa estratégia é muito usada. Já que praticamente todo mundo liga com **Ethernet rápida**, em vez de 802.3u, faremos isso também.

A ideia básica por trás da Ethernet rápida era simples: manter todo o quadro antigo por tapetes, interfaces e regras de procedimento, mas reduzem o tempo de bit de 100 nseg para 10 nsec. Tecnicamente, seria possível copiar a Ethernet clássica de 10 Mbps e ainda detectar colisões a tempo, apenas reduzindo o comprimento máximo do cabo em um fator de 10. No entanto, as vantagens da fiação de par trançado eram tão impressionantes perceber que a Ethernet rápida é inteiramente baseada neste projeto. Assim, todos os sistemas Ethernet rápidos tems usam hubs e switches; cabos multiponto com torneiras vampiro ou conexão BNC toros não são permitidos.

No entanto, algumas escolhas ainda precisavam ser feitas, sendo a mais importante quais tipos de fio apoiar. Um dos competidores era o par trançado da categoria 3. o argumento para isso era que praticamente todos os escritórios no mundo ocidental tinham pelo menos quatro pares trançados de categoria 3 (ou melhor) que vão dele para uma fiação telefônica armário dentro de 100 metros. Às vezes, dois desses cabos existiam. Assim, usando O par trançado de categoria 3 tornaria possível conectar computadores desktop usando Ethernet rápida sem ter que religar o prédio, uma enorme vantagem para muitas organizações.

A principal desvantagem de um par trançado Categoria 3 é sua incapacidade de transportar 100 Mbps em 100 metros, a distância máxima entre o computador e o hub especificada para Hubs de 10 Mbps. Em contraste, a fiação de par trançado Categoria 5 pode lidar com 100 m facilmente, e a fibra pode ir muito mais longe. O compromisso escolhido foi permitir que todos três possibilidades, como mostrado na Fig. 4-19, mas para estimular a solução da Categoria 3 para dar-lhe a capacidade de carga adicional necessária.

O esquema UTP Categoria 3, denominado **100Base-T4**, usava uma velocidade de sinalização de

25 MHz, apenas 25% mais rápido do que 20 MHz da Ethernet padrão. (Lembre-se disso

292

O SUBCAMADA DE CONTROLE DE ACESSO MÉDIO INDIVÍDUO. 4

Nome

Cabo

Máx. segmento

Vantagens

100Base-T4

Par trançado

100 m

Usa UTP de categoria 3

100Base-TX

Par trançado

100 m

Full duplex a 100 Mbps (Cat 5 UTP)

100Base-FX

Fibra ótica

2000 m

Full duplex a 100 Mbps; longas corridas

Figura 4-19. O cabeamento Ethernet rápido original.

Codificação Manchester, discutida na Seç. 2.5, requer dois períodos de relógio para cada dos 10 milhões de bits enviados a cada segundo.) No entanto, para atingir o bit necessário taxa, 100Base-T4 requer quatro pares trançados. Dos quatro pares, um é sempre para o hub, um é sempre do hub e os outros dois são alternáveis para a direção da transmissão atual. Para obter 100 Mbps dos três pares trançados em a direção da transmissão, um esquema bastante complexo é usado em cada par trançado. Envolve o envio de dígitos ternários com três níveis de voltagem diferentes. Este esquema provavelmente não ganhará nenhum prêmio de elegância, e ignoraremos os detalhes. Como desde que a fiação telefônica padrão por décadas teve quatro pares trançados por cabo, a maioria dos escritórios é capaz de usar a instalação elétrica existente. Claro, isso significa desistir do telefone do escritório, mas esse é certamente um pequeno preço a pagar por mais rápido o email.

100Base-T4 caiu no esquecimento porque muitos edifícios de escritórios foram reconectados com UTP categoria 5 para **Ethernet 100Base-TX**, que passou a dominar o mercado.

Este projeto é mais simples porque os fios podem suportar taxas de clock de 125 MHz.

Apenas dois pares trançados por estação são usados, um para o hub e um para ele. Nei- a codificação binária direta (ou seja, NRZ) nem a codificação Manchester é usada. Em vez disso, o **A codificação 4B / 5B** que descrevemos na Seção 2.5 é usada. 4 bits de dados são codificados como 5 sinais

bits finais e enviados a 125 MHz para fornecer 100 Mbps. Este esquema é simples, mas tem transições suficientes para a sincronização e usa a largura de banda do cabo de relação tivamente bem. O sistema 100Base-TX é full duplex; estações podem transmitir em 100 Mbps em um par trançado e recebimento em 100 Mbps em outro par trançado no mesmo tempo.

A última opção, **100Base-FX**, usa dois fios de fibra multimodo, um para cada direção, portanto, também pode executar full duplex com 100 Mbps em cada direção. No esta configuração, a distância entre uma estação e o switch pode ser de até 2 km.

Fast Ethernet permite a interconexão por hubs ou switches. Para garantir que o algoritmo CSMA / CD continua a funcionar, a relação entre o tamanho mínimo da estrutura e o comprimento máximo do cabo devem ser mantidos como a rede a velocidade de trabalho sobe de 10 Mbps para 100 Mbps. Portanto, o quadro mínimo tamanho de 64 bytes deve aumentar ou o comprimento máximo do cabo de 2500 m deve vir para baixo, proporcionalmente. A escolha fácil foi pela distância máxima entre quaisquer duas estações caiam por um fator de 10, uma vez que um hub com cabos de 100 m já está dentro deste novo máximo. No entanto, cabos 100Base-FX de 2 km são

SEC. 4,3
ETHERNET

293

muito longo para permitir um hub de 100 Mbps com o algoritmo de colisão Ethernet normal. Em vez disso, esses cabos devem ser conectados a um switch e operar em full-duplex modo para que não haja colisões.

Os usuários rapidamente começaram a implantar Ethernet rápida, mas eles não estavam prestes a lançar placas Ethernet de 10 Mbps em computadores mais antigos. Como consequência, praticamente todos Switches Ethernet rápidos podem lidar com uma combinação de estações de 10 Mbps e 100 Mbps. Para facilitar a atualização, o próprio padrão fornece um mecanismo chamado **auto-negociação** que permite que duas estações negociem automaticamente a velocidade ótima (10 ou 100 Mbps) e duplexidade (meio ou total). Funciona bem na maioria das vezes, mas é conhecido por levar a problemas de incompatibilidade duplex quando uma extremidade do link autonego-coincide, mas a outra extremidade não e está configurada para o modo full-duplex (Shalunov e Carlson, 2005). A maioria dos produtos Ethernet usa esse recurso para se configurar.

4.3.6 Gigabit Ethernet

A tinta mal secou no padrão Ethernet rápido quando o comitê 802 começou a trabalhar em uma Ethernet ainda mais rápida, rapidamente chamada de **Gigabit Ethernet**. IEEE

ratificou a forma mais popular como 802.3ab em 1999. Abaixo, discutiremos alguns dos os principais recursos da Ethernet gigabit. Mais informações são fornecidas por Spurgeon (2000).

Os objetivos do comitê para Ethernet gigabit eram essencialmente os mesmos que o objetivos do comitê para Ethernet rápida: aumentar o desempenho dez vezes enquanto mantém compatibilidade com todos os padrões Ethernet existentes. Em particular, gigabit Ethernet tinha que oferecer serviço de datagrama não reconhecido com unicast e broadcast, use o mesmo esquema de endereçamento de 48 bits já em uso e mantenha o mesmo formato do quadro, incluindo os tamanhos mínimo e máximo do quadro. O padrão final atingiu todos esses objetivos.

Assim como a Ethernet rápida, todas as configurações de Ethernet gigabit usam ponto a ponto links. Na configuração mais simples, ilustrada na Fig. 4-20 (a), dois computadores são diretamente conectados uns aos outros. O caso mais comum, no entanto, usa um switch ou um hub conectado a vários computadores e possivelmente switches adicionais ou hubs, como mostrado na Fig. 4-20 (b). Em ambas as configurações, cada Ethernet individual o cabo tem exatamente dois dispositivos, nem mais nem menos.

Também como a Ethernet rápida, a Ethernet gigabit suporta dois modos diferentes de operação: modo full-duplex e modo half-duplex. O modo "normal" está completo-modo duplex, que permite o tráfego nas duas direções ao mesmo tempo. Este modo é usado quando há um switch central conectado a computadores (ou outros switches) na periferia. Nesta configuração, todas as linhas são armazenadas em buffer para que cada computador

e o switch é gratuito para enviar frames sempre que quiser. O remetente não tem para detectar o canal para ver se alguém mais o está usando porque a contenção é impossível sible. Na linha entre um computador e um switch, o computador é a única possível remetente para o switch, e a transmissão terá sucesso mesmo se o switch for atualmente enviando um quadro para o computador (porque a linha é full duplex). Desde a

294

O SUBCAMADA DE CONTROLE DE ACESSO MÉDIO
INDIVÍDUO. 4
Switch ou hub
Ethernet

(b)
(uma)
Ethernet
Computador

Figura 4-20. (a) Uma Ethernet de duas estações. (b) Uma Ethernet de várias estações.

nenhuma contenção é possível, o protocolo CSMA / CD não é usado, portanto, o máximo o comprimento do cabo é determinado por problemas de intensidade do sinal, e não por quanto tempo

leva para uma rajada de ruído propagar de volta ao remetente no pior caso.

Switch % es são livres para misturar e combinar velocidades. A negociação automática é suportada apenas

como na Ethernet rápida, só agora a escolha é entre 10, 100 e 1000 Mbps.

O outro modo de operação, half-duplex, é usado quando os computadores estão conectado a um hub em vez de um switch. Um hub não armazena em buffer os quadros recebidos. Em vez disso, ele conecta eletricamente todas as linhas internamente, simulando o multidrop cabo usado na Ethernet clássica. Neste modo, as colisões são possíveis, então o padrão o protocolo dard CSMA / CD é necessário. Porque um quadro de 64 bytes (o mais curto permitido) agora pode ser transmitido 100 vezes mais rápido do que na Ethernet clássica, o o comprimento máximo do cabo deve ser 100 vezes menor, ou 25 metros, para manter o propriedade essencial que o remetente ainda está transmitindo quando a explosão de ruído chega de volta a ele, mesmo no pior dos casos. Com um cabo de 2.500 metros de comprimento, o remetente de um

O quadro de 64 bytes a 1 Gbps seria concluído muito antes de o quadro chegar ao décimo do caminho até a outra extremidade, quanto mais até o fim e voltar.

Esta restrição de comprimento foi dolorosa o suficiente para que dois recursos foram adicionados ao padrão para aumentar o comprimento máximo do cabo para 200 metros, o que é provavelmente o suficiente para a maioria dos escritórios. O primeiro recurso, chamado de **extensão da operadora**, essencialmente

diz ao hardware para adicionar seu próprio preenchimento após o quadro normal para estender o quadro para 512 bytes. Uma vez que este preenchimento é adicionado pelo hardware de envio e removido pelo hardware receptor, o software não tem conhecimento dele, o que significa que nenhum

mudanças são necessárias para o software existente. A desvantagem é que usar 512 bytes de largura de banda para transmitir 46 bytes de dados do usuário (a carga útil de um frame) tem uma eficiência de linha de apenas 9%.

O segundo recurso, chamado **frame bursting**, permite que um remetente transmita uma sequência catenada de vários quadros em uma única transmissão. Se o estouro total for menor que 512 bytes, o hardware o preencherá novamente. Se houver quadros suficientes esperando para transmissão, esse esquema é muito eficiente e preferível à extensão da portadora.

Página 319

SEC. 4,3
ETHERNET

295

Com toda a justiça, é difícil imaginar uma organização comprando computadores modernos com placas Ethernet gigabit e, em seguida, conectá-los a um hub antigo para simular a Ethernet clássica com todas as suas colisões. Interfaces Gigabit Ethernet e os switches costumavam ser caros, mas seus preços caíram rapidamente conforme os volumes de vendas aumentavam

acima. Mesmo assim, a compatibilidade com versões anteriores é sagrada na indústria de computadores,

era necessário um mittee para colocá-lo. Hoje, a maioria dos computadores vem com uma Ethernet interface que é capaz de operação de 10, 100 e 1000 Mbps e compatível com todos eles.

Gigabit Ethernet oferece suporte a cabos de cobre e fibra, conforme listado na Figura 4-21.

A sinalização em ou perto de 1 Gbps requer codificação e envio um pouco a cada nanosegundo. Este truque foi inicialmente realizado com cobre blindado curto

cabos (versão 1000Base-CX) e fibras ópticas. Para as fibras ópticas, dois comprimentos de onda são permitidos e resultam em duas versões diferentes: 0,85 mícrons (curto, para 1000Base-SX) e 1,3 mícrons (longo, para 1000Base-LX).

Nome
Cabo

Máx. segmento

Vantagens

1000Base-SX

Fibra ótica

550 m

Fibra multimodo (50, 62,5 mícrons)

1000Base-LX

Fibra ótica

5000 m

Único (10 μ) ou multimodo (50, 62,5 μ)

1000Base-CX

2 pares de STP

25 m

Par trançado protegido

1000Base-T

4 pares de UTP

100 m

UTP de categoria 5 padrão

Figura 4-21. Cabeamento Gigabit Ethernet.

A sinalização no comprimento de onda curto pode ser obtida com LEDs mais baratos. Isto é usado com fibra multimodo e é útil para conexões dentro de um edifício, pois pode operar até 500 m para fibra de 50 mícrons. Sinalização no comprimento de onda longo requer lasers mais caros. Por outro lado, quando combinado com modo de fibra (10 mícrons), o comprimento do cabo pode ser de até 5 km. Este limite permite longas conexões de distância entre edifícios, como para um backbone de campus, como um link ponto a ponto dedicado. Variações posteriores do padrão permitiram ainda mais links em fibra monomodo.

Para enviar bits por meio dessas versões de Gigabit Ethernet, a codificação **8B / 10B** nós descrito na Seç. 2.5 foi emprestado de outra tecnologia de rede chamada Fibre Channel. Esse esquema codifica 8 bits de dados em palavras-código de 10 bits que são enviados por fio ou fibra, daí o nome 8B / 10B. As palavras-código foram escolhidas para que eles pudessem ser equilibrados (ou seja, ter o mesmo número de 0s e 1s) com suficientes transições fáceis para recuperação do relógio. O envio de bits codificados com NRZ requer uma largura de banda de sinalização de 25% a mais do que a necessária para os bits não codificados, uma grande melhoria em relação à expansão de 100% da codificação Manchester. No entanto, todas essas opções exigiam novos cabos de cobre ou fibra para suportar a sinalização mais rápida. Nenhum deles fez uso da grande quantidade de Categoria 5 UTP que foi instalado junto com a Ethernet rápida. Dentro de um ano, 1000Base-T

1953 frames podem ter se acumulado nessa lacuna. Além disso, quando um computador em um Gigabit Ethernet está enviando dados para um computador em um Ethernet, buffer overruns são muito prováveis. Como consequência dessas duas observações, Gigabit Ethernet suporta controle de fluxo. O mecanismo consiste em um envio final passando um quadro de controle especial para a outra extremidade, dizendo-lhe para pausar por algum período de

Tempo. Esses quadros de controle PAUSE são quadros Ethernet normais contendo um tipo de 0x8808. As pausas são fornecidas em unidades de tempo mínimo de quadro. Para gigabit Ethernet, a unidade de tempo é 512 nseg, permitindo pausas de até 33,6 mseg.

Há mais uma extensão que foi introduzida junto com a Ethernet gigabit.

Os quadros Jumbo permitem que os quadros tenham mais de 1500 bytes, geralmente até 9 KB. Esta extensão é proprietária. Não é reconhecido pelo padrão porque se for usado, então a Ethernet não é mais compatível com as versões anteriores, mas a maioria dos fornecedores

apoia-lo de qualquer maneira. O raciocínio é que 1500 bytes é uma unidade curta de gigabit velocidades. Ao manipular blocos maiores de informações, a taxa de quadros pode ser diminuído, junto com o processamento associado a ele, como interromper o processador para dizer que um quadro chegou, ou dividir e recombinar a mensagem sábios que eram muito longos para caber em um quadro Ethernet.

4.3.7 Ethernet 10-Gigabit

Assim que o gigabit Ethernet foi padronizado, o comitê 802 ficou entediado e queria voltar ao trabalho. O IEEE disse a eles para começarem com Ethernet de 10 gigabits. Este trabalho seguiu o mesmo padrão dos padrões Ethernet anteriores, com padrões para fibra e cabo de cobre blindado aparecendo pela primeira vez em 2002 e 2004, seguido pelo padrão para par trançado de cobre em 2006.

10 Gbps é uma velocidade verdadeiramente prodigiosa, 1000 vezes mais rápida que a Ethernet original.

Onde poderia ser necessário? A resposta está dentro de data centers e trocas para conectar roteadores, switches e servidores de última geração, bem como de longa distância, alta troncos de largura de banda entre escritórios que habilitam toda a rede da área metropolitana funciona com base em Ethernet e fibra. As conexões de longa distância usam óptica fibra, enquanto as conexões curtas podem usar cobre ou fibra.

Página 321

SEC. 4,3
ETHERNET

297

Todas as versões de Ethernet de 10 gigabits suportam apenas operação full-duplex. CSMA / CD não faz mais parte do design e os padrões se concentram no detalhes de camadas físicas que podem ser executadas em alta velocidade. Compatibilidade ainda importa, no entanto, as interfaces Ethernet de 10 gigabits são autonegociadas e voltam para a velocidade mais alta suportada por ambas as extremidades da linha.

Os principais tipos de Ethernet de 10 gigabits estão listados na Figura 4-22. Multimodo fibra com comprimento de onda de 0,85 μ (curto) é usada para distâncias médias e a fibra modal em 1,3 μ (longo) e 1,5 μ (estendido) é usada para longas distâncias.

10GBase-ER pode percorrer distâncias de 40 km, tornando-o adequado para áreas extensas formulários. Todas essas versões enviam um fluxo serial de informações que são produzida pela codificação dos bits de dados e, em seguida, codificando-os com um código **64B / 66B**. Isto

a codificação tem menos sobrecarga do que um código 8B / 10B.

Nome

Cabo

Máx. segmento

Vantagens

10GBase-SR

Fibra ótica

Até 300 m

Fibra multimodo (0,85 μ)

10GBase-LR
Fibra ótica
10 km
Fibra monomodo ($1,3 \mu$)
10GBase-ER
Fibra ótica
40 km
Fibra monomodo ($1,5 \mu$)
10GBase-CX4
4 pares de twinax
15 m
Cobre biaxial
10GBase-T
4 pares de UTP
100 m
Categoria 6a UTP

Figura 4-22. Cabeamento Ethernet de 10 Gigabit.

A primeira versão de cobre definida, 10GBase-CX4, usa um cabo com quatro pares de fiação de cobre biaxial. Cada par usa codificação 8B / 10B e executa em 3,125 Símbolos Gs / segundo para atingir 10 Gbps. Esta versão é mais barata que a fibra e era cedo para o mercado, mas resta saber se será superado no longo prazo executado por Ethernet de 10 gigabits em uma variedade de fiação de par trançado. 10GBase-T é uma versão que usa cabos UTP. Embora exija a Categoria 6a fiação, para execuções mais curtas, pode usar categorias inferiores (incluindo a Categoria 5) para permitir alguma reutilização do cabeamento instalado. Não surpreendentemente, a camada física é bastante envolvida para alcançar 10 Gbps em par trançado. Vamos apenas esboçar alguns dos detalhes de alto nível. Cada um dos quatro pares trançados é usado para enviar 2500 Mbps em ambas direções. Esta velocidade é alcançada usando uma taxa de sinalização de 800 Msymbols / seg com símbolos que usam 16 níveis de tensão. Os símbolos são produzidos por embaralhamento os dados, protegendo-os com um código LDPC (Low Density Parity Check) e mais codificação para correção de erros. A Ethernet de 10 gigabits ainda está tremendo no mercado, mas o comitê 802.3 já mudou. No final de 2007, o IEEE criou um grupo para padronizar Ethernet operando a 40 Gbps e 100 Gbps. Esta atualização permitirá que a Ethernet compete em configurações de alto desempenho, incluindo conexões de longa distância em redes de backbone e conexões curtas sobre os backplanes dos equipamentos. O padrão ainda não está completo, mas produtos proprietários já estão disponíveis.

298

O SUBCAMADA DE CONTROLE DE ACESSO MÉDIO INDIVÍDUO. 4

4.3.8 Retrospectiva na Ethernet

Ethernet existe há mais de 30 anos e não tem concorrentes sérios em vista, por isso é provável que continue por muitos anos. Poucas arquiteturas de CPU, sistemas operacionais ou linguagens de programação têm sido o rei da montanha por três décadas continuando fortes. Claramente, a Ethernet fez algo certo. O que? Provavelmente, a principal razão de sua longevidade é que a Ethernet é simples e flexível. Na prática, simples se traduz em confiável, barato e fácil de manter. Uma vez a arquitetura de hub e switch foi adotada, as falhas tornaram-se extremamente raras. As pessoas hesitam em substituir algo que funciona perfeitamente o tempo todo, especialmente quando sabem que muitas coisas na indústria de computadores funcionam muito mal, de modo que muitos dos chamados "upgrades" são piores do que o que substituíram. Simples também se traduz em barato. A fiação de par trançado é relativamente barata sive como são os componentes de hardware. Eles podem começar caros quando há uma transição, por exemplo, novos NICs ou switches Gigabit Ethernet, mas eles são meramente adições a uma rede bem estabelecida (não uma substituição dela) e os preços caem rapidamente à medida que o volume de vendas aumenta. Ethernet é fácil de manter. Não há software para instalar (exceto o

drivers) e não muito na forma de tabelas de configuração para gerenciar (e obter errado). Além disso, adicionar novos hosts é tão simples quanto conectá-los. Outro ponto é que a Ethernet funciona facilmente com TCP / IP, que tem tornar-se dominante. IP é um protocolo sem conexão, por isso se encaixa perfeitamente com Ethernet, que também não tem conexão. IP se encaixa muito menos bem com orientado a conexão alternativas como ATM. Essa incompatibilidade definitivamente prejudicou as chances do ATM. Por último, e talvez o mais importante, a Ethernet foi capaz de evoluir em certas maneiras cruciais. As velocidades aumentaram em várias ordens de magnitude e hubs e interruptores foram introduzidos, mas essas mudanças não exigiram mudanças no software e muitas vezes permitem que o cabeamento existente seja reutilizado por um tempo. Quando um vendedor de rede aparece em uma grande instalação e diz "Eu tenho isso fantástica nova rede para você. Tudo que você precisa fazer é descartar todo o seu hardware e reescrever todo o seu software, " ele tem um problema. Muitas tecnologias alternativas das quais você provavelmente nem ouviu falar eram mais rápido do que a Ethernet quando foram introduzidos. Bem como ATM, esta lista inclui FDDI (Interface de dados distribuída de fibra) e Fibre Channel,
†
dois anéis- LANs ópticas baseadas. Ambos eram incompatíveis com Ethernet. Nenhum deles conseguiu. Eles eram muito complicados, o que gerava chips complexos e preços altos. O less-filho que deveria ter sido aprendido aqui era KISS (Keep It Simple, Stupid). Até finalmente, a Ethernet os alcançou em termos de velocidade, muitas vezes tomando emprestado alguns de sua tecnologia, por exemplo, a codificação 4B / 5B da FDDI e a codificação 8B / 10B codificação do Fibre Channel. Então eles não tiveram mais vantagens e silenciosamente morreram ou caiu em funções especializadas.
† É denominado " Fibre Channel " e não " Fibre Channel " porque o editor de documentos era britânico.

Página 323

SEC. 4,3
ETHERNET

299

Parece que a Ethernet continuará a se expandir em seus aplicativos para alguns Tempo. A Ethernet de 10 gigabits o livrou das restrições de distância do CSMA / CD. Muito esforço está sendo colocado em **Ethernet de nível de operadora** para permitir que os provedores de rede ofereçam serviços baseados em Ethernet aos seus clientes para áreas metropolitanas e extensas redes (Fouli e Maler, 2009). Este aplicativo carrega quadros Ethernet longos distâncias sobre fibra e chamadas para melhores recursos de gerenciamento para ajudar as operadoras oferecer serviços confiáveis e de alta qualidade. Redes de alta velocidade também estão encontrando uso em backplanes conectando componentes em grandes roteadores ou servidores. Ambos esses usos são adicionais ao envio de frames entre computadores em escritórios.

4.4 LANS SEM FIO

LANs sem fio são cada vez mais populares, e residências, escritórios, cafés, bibliotecas, aeroportos, zoológicos e outros locais públicos estão sendo equipados com eles para se conectar computadores, PDAs e telefones inteligentes para a Internet. LANs sem fio também podem ser usado para permitir que dois ou mais computadores próximos se comuniquem sem usar o Internet.

O principal padrão de LAN sem fio é 802.11. Fornecemos algumas informações básicas informações sobre ele na Seç. 1.5.3. Agora é hora de examinar mais de perto a tecnologia. Nas seções a seguir, veremos a pilha de protocolos, o rádio da camada física técnicas de transmissão, o protocolo de subcamada MAC, a estrutura do quadro e os serviços prestados. Para obter mais informações sobre 802.11, consulte Gast (2005). Para obter a verdade da boca do cavalo, consulte a norma publicada, IEEE 802.11-2007 em si.

4.4.1 A arquitetura 802.11 e pilha de protocolo

As redes 802.11 podem ser usadas em dois modos. O modo mais popular é conectar clientes, como laptops e smartphones, a outra rede, como um computador intranet da empresa ou na Internet. Esse modo é mostrado na Figura 4-23 (a). Em infraestrutura modo, cada cliente está associado a um **AP** (**Ponto de Acesso**) que, por sua vez, conectado à outra rede. O cliente envia e recebe seus pacotes através do AP.

Vários pontos de acesso podem ser conectados juntos, normalmente por uma rede com fio chamado de **sistema de distribuição**, para formar uma rede 802.11 estendida. Nesse caso, os clientes podem enviar frames a outros clientes por meio de seus APs.

O outro modo, mostrado na Figura 4.23 (b), é uma **rede ad hoc**. Este modo é uma coleção de computadores que estão associados para que possam enviar quadros diretamente para entre si. Não há ponto de acesso. Como o acesso à Internet é o aplicativo matador para redes sem fio, ad hoc não são muito populares.

Agora veremos os protocolos. Todos os protocolos 802, incluindo 802.11 e Ethernet, têm uma certa estrutura comum. Uma visão parcial do

A pilha do protocolo 802.11 é apresentada na Figura 4.24. A pilha é a mesma para clientes e

Página 324

300

O SUBCAMADA DE CONTROLE DE ACESSO MÉDIO INDIVÍDUO. 4

(uma)

(b)

Para rede

Acesso

ponto

Cliente

Figura 4-23. Arquitetura 802.11. (a) Modo de infraestrutura. (b) Modo Ad-hoc.

APs. A camada física corresponde bastante bem à camada física OSI, mas o

A camada de enlace de dados em todos os protocolos 802 é dividida em duas ou mais subcamadas. No

802.11, a subcamada MAC (Controle de Acesso Médio) determina como o canal é alocado, ou seja, quem transmite em seguida. Acima está o LLC (Link Lógico Control) subcamada, cujo trabalho é esconder as diferenças entre os diferentes 802 variantes e torná-los indistinguíveis no que diz respeito à camada de rede.

Isso poderia ter sido uma responsabilidade significativa, mas hoje em dia a LLC é uma cola camada que identifica o protocolo (por exemplo, IP) que é transportado dentro de um quadro 802.11.

802.11 (legado)

Freqüência

saltar

e infravermelho

802.11a

OFDM

802.11b

Propagação

espectro

802.11g

OFDM

802.11n

MIMO

OFDM

Camada de link lógico

Data de lançamento:

1997–1999

1999

1999

2003

2009

Superior

camadas

Link de dados

camada

Física

camada

MAC

subcamada

Figura 4-24. Parte da pilha do protocolo 802.11.

Várias técnicas de transmissão foram adicionadas à camada física como

802.11 evoluiu desde que apareceu pela primeira vez em 1997. Duas das técnicas iniciais, infravermelho na forma de controles remotos de televisão e salto de frequência no

Banda de 2,4 GHz, agora estão extintas. A terceira técnica inicial, sequência direta espalhar espectro a 1 ou 2 Mbps na banda de 2,4 GHz, foi estendido para funcionar a taxas até 11 Mbps e rapidamente se tornou um sucesso. Agora é conhecido como 802.11b.

Página 325

SEC. 4,4
LANS SEM FIO

301

Para dar aos viciados em wireless um aumento de velocidade muito desejado, uma nova tecnologia de transmissão niques baseados em OFDM (Multiplexação por Divisão de Frequência Ortogonal) esquema que descrevemos na Seç. 2.5.3 foram introduzidos em 1999 e 2003. O primeiro é chamado 802.11ae usa uma banda de frequência diferente, 5 GHz. O segundo ficou com 2,4 GHz e compatibilidade. É denominado 802.11g. Ambos fornecem taxas de até 54 Mbps. Mais recentemente, as técnicas de transmissão que usam simultaneamente vários tennas no transmissor e receptor para um aumento de velocidade foram finalizados como 802.11n em outubro de 2009. Com quatro antenas e canais mais amplos, o padrão 802.11 agora define taxas de até 600 Mbps.

Vamos agora examinar cada uma dessas técnicas de transmissão brevemente. Nós vamos cobrem apenas aqueles que estão em uso, no entanto, ignorando a transmissão 802.11 legada métodos. Tecnicamente, eles pertencem à camada física e deveriam ter sido examinado no cap. 2, mas uma vez que eles estão tão intimamente ligados às LANs em geral e ao 802.11 LAN em particular, nós os tratamos aqui.

4.4.2 A Camada Física 802.11

Cada uma das técnicas de transmissão torna possível enviar um quadro MAC pelo ar de uma estação para outra. Eles diferem, no entanto, na tecnologia usado e velocidades alcançáveis. Uma discussão detalhada dessas tecnologias está longe além do escopo deste livro, mas algumas palavras sobre cada um irão relacionar a técnica questões ao material que cobrimos na Seç. 2.5 e fornecerá leitores interessados com os termos-chave para pesquisar em outro lugar para obter mais informações. Todas as técnicas 802.11 usam rádios de curto alcance para transmitir sinais em ei- as bandas de frequência ISM de 2,4 GHz ou 5 GHz, ambas descritas na Seç. 2.3.3. Essas bandas têm a vantagem de não serem licenciadas e, portanto, estão disponíveis gratuitamente para qualquer transmissor disposto a atender a algumas restrições, como potência irradiada de a maioria 1 W (embora 50 mW seja mais comum para rádios LAN sem fio). Infeliz- ly, este fato também é conhecido pelos fabricantes de abridores de porta de garagem, sem fio telefones, fornos de microondas e inúmeros outros dispositivos, todos os quais competem com laptops para o mesmo espectro. A banda de 2,4 GHz tende a ser mais lotada do que a banda de 5 GHz, então 5 GHz pode ser melhor para alguns aplicativos, embora tenha intervalo mais curto devido à frequência mais alta.

Todos os métodos de transmissão também definem taxas múltiplas. A ideia é que taxas diferentes podem ser usadas dependendo das condições atuais. Se o wireless o sinal está fraco, uma taxa baixa pode ser usada. Se o sinal estiver claro, a taxa mais alta pode ser usava. Este ajuste é chamado de **adaptação de taxa**. Uma vez que as taxas variam por um fator de 10 ou mais, uma boa adaptação de taxa é importante para um bom desempenho. Claro, uma vez que não é necessário para a interoperabilidade, os padrões não dizem como a taxa de adaptação deve ser feita.

O primeiro método de transmissão que veremos é **802.11b**. É um spread-spec- método trum que suporta taxas de 1, 2, 5,5 e 11 Mbps, embora na prática o a taxa de operação é quase sempre 11 Mbps. É semelhante ao sistema CDMA que nós

Página 326

302

O SUBCAMADA DE CONTROLE DE ACESSO MÉDIO

INDIVÍDUO. 4

examinado na Seç. 2.5, exceto que há apenas um código de propagação que é compartilhado por todos os usuários. A distribuição é usada para satisfazer o requisito da FCC de que o poder seja espalhados pela banda ISM. A sequência de espalhamento usada por 201.11b é um **Barker sequência**. Ele tem a propriedade de que sua autocorrelação seja baixa, exceto quando o as respostas estão alinhadas. Esta propriedade permite que um receptor trave no início de um transmissão. Para enviar a uma taxa de 1 Mbps, a sequência Barker é usada com Modulação BPSK para enviar 1 bit por 11 chips. Os chips são transmitidos a uma taxa de 11 Mchips / seg. Para enviar a 2 Mbps, é usado com modulação QPSK para enviar 2 bits por 11 chips. As taxas mais altas são diferentes. Essas taxas usam uma técnica chamada-**CCK (Complementary Code Keying)** para construir códigos em vez do Sequência de Barker. A taxa de 5,5 Mbps envia 4 bits em cada código de 8 chips, e o A taxa de 11 Mbps envia 8 bits em cada código de 8 chips.

Em seguida, chegamos ao **802.11a**, que suporta taxas de até 54 Mbps na faixa de 5 GHz Banda ISM. Você poderia esperar que 802.11a viesse antes de 802.11b, mas Esse não foi o caso. Embora o grupo 802.11a tenha sido configurado primeiro, o 802.11b padrão foi aprovado primeiro e seu produto chegou ao mercado bem à frente do Produtos 802.11a, em parte devido à dificuldade de operar nos 5 GHz superiores banda.

O método 802.11a é baseado em **OFDM (Divisão de Freqüência Ortogonal Multiplexação)** porque OFDM usa o espectro de forma eficiente e resiste a redes sem fio degradações de sinal, como multipath. Os bits são enviados para 52 subportadoras em paralelo lel, 48 transportando dados e 4 usados para sincronização. Cada símbolo dura $4\mu s$ e envia 1, 2, 4 ou 6 bits. Os bits são codificados para correção de erros com um binário com código volucional primeiro, portanto, apenas 1/2, 2/3 ou 3/4 dos bits não são redundantes. Com combinações diferentes, 802.11a pode ser executado em oito taxas diferentes, variando de 6 a 54 Mbps. Essas taxas são significativamente mais rápidas do que as taxas 802.11b, e há menos interferência na banda de 5 GHz. No entanto, 802.11b tem um intervalo de cerca de sete vezes maior do que 802.11a, que é mais importante em muitas situações.

Mesmo com o alcance maior, as pessoas do 802.11b não tinham intenção de permitir este novato vence o campeonato de velocidade. Felizmente, em maio de 2002, o FCC abandonou sua regra de longa data exigindo todos os equipamentos de comunicação sem fio operando nas bandas ISM nos EUA para usar espectro de dispersão, então começou a funcionar em **802.11g**, que foi aprovado pelo IEEE em 2003. Ele copia os módulos OFDM- métodos de instalação de 802.11a, mas opera na banda ISM estreita de 2,4 GHz junto com 802.11b. Ele oferece as mesmas taxas de 802.11a (6 a 54 Mbps) mais, é claro, comparação com qualquer dispositivo 802.11b que esteja por perto. Todos esses diferentes as escolhas podem ser confusas para os clientes, por isso é comum os produtos oferecerem suporte 802.11a / b / g em um único NIC.

Não contente em parar por aí, o comitê do IEEE começou a trabalhar em uma colocar camada física chamada **802.11n**. Foi ratificado em 2009. A meta para 802.11n foi a taxa de transferência de pelo menos 100 Mbps depois que todas as sobrecargas sem fio foram re- mudou-se. Essa meta exigia um aumento bruto de velocidade de pelo menos um fator de quatro. Para fazer acontecer, o comitê dobrou os canais de 20 MHz para 40 MHz e

SEC. 4,4
LANS SEM FIO

303

despesas gerais de enquadramento reduzidas, permitindo que um grupo de quadros seja enviado em conjunto.

Mais significativamente, no entanto, 802.11n usa até quatro antenas para transmitir até quatro fluxos de informação ao mesmo tempo. Os sinais dos streams interferem no receptor, mas eles podem ser separados usando **MIMO (Multiple Input Multiple Saída)** técnicas de comunicação. O uso de várias antenas oferece uma grande aumento de velocidade ou melhor alcance e confiabilidade. MIMO, como OFDM, é um dos

aquelas ideias de comunicação inteligentes que estão mudando os designs sem fio e que provavelmente ouviremos muito sobre isso no futuro. Para uma breve introdução ao multi-antenas em 802.11 ver Halperin et al. (2010).

4.4.3 O protocolo de subcamada 802.11 MAC

Vamos agora retornar da terra da engenharia elétrica para a terra da comp. ciênciencia do computador. O protocolo de subcamada 802.11 MAC é bastante diferente daquele de Ethernet, devido a dois fatores que são fundamentais para a comunicação sem fio.

Em primeiro lugar, os rádios são quase sempre half duplex, o que significa que eles não podem transmitir

e escute rajadas de ruído ao mesmo tempo em uma única frequência. O recebido o sinal pode ser facilmente um milhão de vezes mais fraco do que o sinal transmitido, então pode não ser ouvido ao mesmo tempo. Com Ethernet, uma estação apenas espera até o éter fica em silêncio e começa a transmitir. Se não receber um ruído de retorno ao transmitir os primeiros 64 bytes, o quadro quase com certeza foi entregue corretamente. Com wireless, este mecanismo de detecção de colisão não funciona.

Em vez disso, 802.11 tenta evitar colisões com um protocolo chamado **CSMA / CA** (**CSMA com prevenção de colisão**). Este protocolo é conceitualmente semelhante a CSMA / CD da Ethernet, com detecção de canal antes do envio e retorno exponencial fora após colisões. No entanto, uma estação que tem um quadro para enviar começa com um dom backoff (exceto no caso de não ter usado o canal recentemente e o canal está ocioso). Não espera por uma colisão. O número de slots para retirada é escolhido no intervalo de 0 a, digamos, 15 no caso da camada física OFDM. O estação espera até que o canal esteja ocioso, sentindo que não há sinal por um curto período de tempo (chamado de DIFS, como explicamos abaixo) e faz a contagem regressiva dos slots ociosos,

pausando quando os quadros são enviados. Ele envia seu quadro quando o contador chega a 0. Se o quadro é transmitido, o destino imediatamente envia uma breve confirmação. A falta de um reconhecimento é inferida para indicar um erro, seja uma colisão ou não. Neste caso, o remetente dobra o período de retirada e tenta novamente, continuando com backoff exponencial como na Ethernet até que o quadro tenha sido transmitido com sucesso ou o número máximo de retransmissões foi alcançado.

Um exemplo de linha do tempo é mostrado na Figura 4-25. A estação A é a primeira a enviar um quadro, Armação. Enquanto A está enviando, as estações B e C ficam prontas para enviar. Eles veem isso

o canal está ocupado e aguarde até que ele fique ocioso. Pouco depois de A receber um ac-
conhecimento, o canal fica ocioso. No entanto, em vez de enviar um quadro
imediatamente e colidindo, B e C realizam um recuo. C pega um pequeno recuo,

Página 328

304

O SUBCAMADA DE CONTROLE DE ACESSO MÉDIO INDIVÍDUO. 4

e, portanto, envia primeiro. B pausa sua contagem regressiva enquanto sente que C está usando o canal, e continua após C receber uma confirmação. Em breve com preenche seu backoff e envia seu quadro.

Estação
UMA
B
C
Tempo
Dados
Espere pelo Backoff ocioso
Resto do recuo
Ack
A envia para D
B pronto para enviar
D acks A
C envia para D
D acks C
B envia para D
D acks B

```

Dados
Ack
Dados
Ack
Espere por inativo
Espere pelo Backoff ocioso
C pronto para enviar

```

Figura 4-25. Enviando um quadro com CSMA / CA.

Em comparação com a Ethernet, existem duas diferenças principais. Primeiro, começando backoffs cedo ajuda a evitar colisões. Essa evitação vale a pena porque as colisões são caros, pois todo o quadro é transmitido, mesmo que ocorra um. Em segundo lugar, conhecimentos são usados para inferir colisões porque as colisões não podem ser detectadas. Este modo de operação é denominado **DCF (Função de Coordenação Distribuída)** porque cada estação atua de forma independente, sem nenhum tipo de controle central. O padrão também inclui um modo opcional de operação chamado **PCF (Point Coordinação de nação)** em que o ponto de acesso controla todas as atividades em sua célula, apenas como uma estação base de celular. No entanto, o PCF não é usado na prática porque há normalmente nenhuma maneira de evitar que estações em outra rede próxima transmitam tráfego concorrente.

O segundo problema é que as faixas de transmissão de diferentes estações podem ser diferente. Com um fio, o sistema é projetado para que todas as estações possam ouvir entre si. Com as complexidades da propagação de RF, esta situação não se mantém para estações sem fio. Consequentemente, situações como o terminal oculto problema mencionado anteriormente e ilustrado novamente na Figura 4.26 (a) pode surgir. Uma vez que nem todos as estações estão dentro do alcance de rádio umas das outras, as transmissões acontecendo em uma parte da célula não pode ser recebida em outro lugar na mesma célula. Neste exemplo, estação C está a transmitir para a estação B. Se A sentir o canal, não ouvirá nada e falsamente concluir que ele pode agora começar a transmitir para B. Esta decisão leva para uma colisão.

A situação inversa é o problema do terminal exposto, ilustrado na Fig. 4-26 (b). Aqui, B deseja enviar para C, para que ouça o canal. Quando ouve um

SEC. 4,4
LANS SEM FIO

305

```

Alcance
de C's
rádio
UMA
C
B
(uma)
UMA
C
Alcance
como de
rádio
B
(b)
A quer enviar para B
mas não posso ouvir isso
B está ocupado
B quer enviar para C
mas pensa erroneamente
a transmissão irá falhar
C é
transmitindo
A é
transmitindo

```

Figura 4-26. (a) O problema do terminal oculto. (b) O problema do terminal exposto.

transmissão, ele conclui falsamente que não pode enviar para C, mesmo que A possa em fato estar transmitindo para D (não mostrado). Esta decisão desperdiça uma oportunidade de transmissão afinidade.

Para reduzir ambigüidades sobre qual estação está enviando, 802.11 define o canal de detecção consiste em detecção física e detecção virtual. Detecção física simplesmente verifica o meio para ver se há um sinal válido. Com sensor virtual, cada estação mantém um registro lógico de quando o canal está em uso, rastreando o NAV (**Vetor de alocação de rede**). Cada quadro carrega um campo NAV que diz quanto tempo a sequência da qual este quadro faz parte levará para ser concluída. Estações que ouvirem este quadro saberão que o canal estará ocupado pelo período identificados pelo NAV , independentemente de poderem detectar um sinal físico. Para exemplo, o NAV de um quadro de dados inclui o tempo necessário para enviar uma confirmação. Todas as estações que ouvem o quadro de dados irão adiar durante o reconhecimento período, se eles podem ouvir ou não o reconhecimento.

Um mecanismo RTS / CTS opcional usa o NAV para evitar que os terminais envio de frames ao mesmo tempo que terminais ocultos. Isso é mostrado na Fig. 4-27. No Neste exemplo, *A* quer enviar para *B* . *C* é uma estação dentro do alcance de *A* (e possivelmente dentro do intervalo de *B* , mas isso não importa). *D* é uma estação dentro do alcance de *B*, mas não dentro do alcance de *um* .

O protocolo começa quando *um* decide que quer enviar dados para *B* . *A* começa por enviar um quadro RTS para *B* para solicitar permissão para enviar um quadro. Se *B* receber esta solicitação, ele responde com um quadro CTS para indicar que o canal está livre para enviar. Ao receber o CTS , *A* envia seu quadro e inicia um temporizador ACK . Sobre recebimento correto do quadro de dados, *B* responde com um quadro ACK , completando o troca. Se *A* 's ACK temporizador expira antes do ACK recebe de volta a ele, ele é tratado como uma colisão e todo o protocolo é executado novamente após um recuo.

Página 330

306

O SUBCAMADA DE CONTROLE DE ACESSO MÉDIO INDIVÍDUO. 4

RTS
Dados
UMA
CTS
ACK
B
C
D
NAV
NAV
Tempo

Figura 4-27. Sensor de canal virtual usando CSMA / CA.

Agora vamos considerar essa troca de pontos de vista de *C* e *D* . *C* está com na faixa de *A* , para que possa receber o quadro RTS . Se isso acontecer, ele percebe que alguém vai enviar dados em breve. A partir das informações fornecidas na solicitação RTS , pode estimar quanto tempo a sequência levará, incluindo o ACK final . Então, para o bom de tudo, ele desiste de transmitir qualquer coisa até que a troca seja concluída. Ele faz isso atualizando seu registro do NAV para indicar que o canal está ocupado, como mostrado na Fig. 4-27. *D* não ouve o RTS , mas ouve o CTS , portanto, também atualiza seu NAV . Observe que os sinais NAV não são transmitidos; eles estão apenas dentro lembretes ternais para ficar quieto por um certo período de tempo.

No entanto, embora RTS / CTS pareça bom em teoria, é um daqueles projetos que provou ser de pouco valor na prática. Vários motivos pelos quais raramente é usado são conhecidos. Não ajuda para quadros curtos (que são enviados no lugar do RTS) ou para o AP (que todos podem ouvir, por definição). Para outras situações, apenas retarda a operação. RTS / CTS em 802.11 é um pouco diferente do que no Protocolo MACA que vimos na Seção 4.2 porque todos que ouvem o RTS ou CTS permanece quieto durante todo o tempo para permitir que o ACK passe sem colisão. Por causa disso, não ajuda com terminais expostos como MACA fez, apenas com terminais ocultos. Na maioria das vezes, existem alguns terminais ocultos, e CSMA / CA al-pronto os ajuda reduzindo a velocidade de estações que transmitem sem sucesso, seja o que for a causa, para tornar mais provável o sucesso das transmissões.

CSMA / CA com detecção física e virtual é o núcleo do proto 802.11

col. No entanto, existem vários outros mecanismos que foram desenvolvidos para ir com isso. Cada um desses mecanismos foi impulsionado pelas necessidades de operação real, então vamos examiná-los brevemente.

A primeira necessidade que examinaremos é a confiabilidade. Em contraste com as redes com fio, redes sem fio são barulhentas e não confiáveis, em grande parte devido à interferência de outros tipos de dispositivos, como fornos de microondas, que também usam o bandas ISM censuradas. O uso de reconhecimentos e retransmissões é de pouco ajuda se a probabilidade de obter um quadro é pequena em primeiro lugar.

Página 331

SEC. 4,4
LANS SEM FIO
307

A principal estratégia usada para aumentar as transmissões bem-sucedidas é diminuir a taxa de transmissão. Taxas mais lentas usam modulações mais robustas que são mais provável de ser recebido corretamente para uma determinada relação sinal-ruído. Se muitos quadros são perdidos, uma estação pode diminuir a taxa. Se os quadros forem entregues com pouco perda, uma estação pode ocasionalmente testar uma taxa mais alta para ver se ela deve ser usada.

Outra estratégia para melhorar a chance de o quadro passar por undam-envelhecido é enviar quadros mais curtos. Se a probabilidade de qualquer bit estar em erro é p , a probabilidade de um quadro de n bits ser recebido de forma totalmente correta é $(1 - p)^n$. Para exemplo, para $p = 10^{-4}$

, a probabilidade de receber um quadro Ethernet completo (12.144 bits) corretamente é inferior a 30%. A maioria dos quadros será perdida. Mas se as molduras forem apenas um terço (4048 bits), dois terços deles serão recebidos corretamente. Agora a maioria dos quadros será transmitida e menos retransmissões serão necessárias.

Quadros mais curtos podem ser implementados reduzindo o tamanho máximo do mensagem que é aceita da camada de rede. Como alternativa, 802.11 permite quadros a serem divididos em pedaços menores, chamados **fragmentos**, cada um com sua própria verificação

soma. O tamanho do fragmento não é fixado pelo padrão, mas é um parâmetro que pode ser ajustado pela AP. Os fragmentos são numerados individualmente e reconhecidos usando um protocolo de parar e esperar (ou seja, o remetente não pode transmitir o fragmento $k + 1$ até que tenha recebido a confirmação do fragmento k). Assim que o canal tiver sido adquirido, vários fragmentos são enviados como uma explosão. Eles vão um após o outro com uma confirmação (e possivelmente retransmissões) no meio, até que todo o quadro foi enviado com sucesso ou o tempo de transmissão atinge o máximo permitido. O mecanismo NAV mantém outras estações silenciosas apenas até o próxima confirmação, mas outro mecanismo (veja abaixo) é usado para permitir um rajada de fragmentos a serem enviados sem que outras estações enviem um quadro no meio.

A segunda necessidade que discutiremos é a economia de energia. A vida da bateria é sempre um problema com dispositivos sem fio móveis. O padrão 802.11 presta atenção ao questão de gerenciamento de energia para que os clientes não precisem desperdiçar energia quando tiverem

nem informação para enviar nem receber.

O mecanismo básico para economizar energia é baseado em **estruturas de farol**. Beacons são transmissões periódicas pelo AP (por exemplo, a cada 100 ms). Os quadros anunciam a presença do AP para clientes e transportar parâmetros do sistema, como o identificador fier do AP, o tempo, quanto tempo até o próximo beacon e configurações de segurança.

Os clientes podem definir um bit de gerenciamento de energia nos quadros que eles enviam ao AP para

diga que eles estão entrando **no modo de economia de energia**. Neste modo, o cliente pode cochilar

e o AP armazenará em buffer o tráfego destinado a ele. Para verificar o tráfego de entrada, o cliente acorda para cada beacon e verifica um mapa de tráfego que é enviado como parte de o farol. Este mapa informa ao cliente se há tráfego em buffer. Se sim, o cliente envia uma mensagem de pesquisa ao AP, que então envia o tráfego no buffer. O cliente

pode então voltar a dormir até que o próximo beacon seja enviado.

Outro mecanismo de economia de energia, chamado **APSD** (**A**utomatic **P**ower **S**ave **D**elivery), também foi adicionado ao 802.11 em 2005. Com esse novo mecanismo, o AP armazena quadros em buffer e os envia para um cliente logo após o cliente enviar quadros para o

308

O SUBCAMADA DE CONTROLE DE ACESSO MÉDIO INDIVÍDUO. 4

AP. O cliente pode então ir dormir até que tenha mais tráfego para enviar (e receber).

Este mecanismo funciona bem para aplicações como VoIP que têm tráfego frequente fic em ambas as direções. Por exemplo, um telefone sem fio VoIP pode usá-lo para enviar e receber quadros a cada 20 ms, com muito mais frequência do que o intervalo do beacon de 100 ms, enquanto cochila no meio.

A terceira e última necessidade que examinaremos é a qualidade do serviço. Quando o VoIP tráfego no exemplo anterior compete com o tráfego ponto a ponto, o tráfego de VoIP fic vai sofrer. Será atrasado devido à contenção com a alta largura de banda tráfego ponto a ponto, mesmo que a largura de banda VoIP seja baixa. Esses atrasos são susceptível de degradar as chamadas de voz. Para evitar essa degradação, gostaríamos de deixar o tráfego VoIP vai à frente do tráfego ponto a ponto, pois é de maior prioridade.

IEEE 802.11 tem um mecanismo inteligente para fornecer este tipo de qualidade de serviço que foi introduzido como um conjunto de extensões sob o nome 802.11e em 2005. Funciona estendendo CSMA / CA com intervalos cuidadosamente definidos entre os quadros. Após um quadro foi enviado, uma certa quantidade de tempo ocioso é necessária antes de qualquer estação pode enviar um quadro para verificar se o canal não está mais em uso. O truque é definir diferentes intervalos de tempo para diferentes tipos de quadros.

Cinco intervalos são representados na Fig. 4-28. O intervalo entre os dados regulares frames é chamado de **DIFS** (**D**CF **I**nter**F**rame **S**pace). Qualquer estação pode tentar adquirir o canal para enviar um novo quadro depois que o meio estiver inativo por DIFS. As regras de contenção usuais se aplicam, e o backoff exponencial binário pode ser necessário se ocorrer uma colisão. O intervalo mais curto é **SIFS** (**S**hort **I**nter**F**rame **E**spaçamento). É usado para permitir às partes em um único diálogo a chance de serem os primeiros.

Os exemplos incluem permitir que o receptor envie um ACK , outras sequências de quadro de controle

como RTS e CTS , ou permitir que um remetente transmita uma rajada de fragmentos. Enviando o próximo fragmento após esperar apenas SIFS é o que impede outra estação de saltar entrando com uma moldura no meio da troca.

ACK

SIFS

AIFS 1

DIFS

EIFS

AIFS 4

Quadro de controle ou próximo fragmento pode ser enviado aqui

Quadro de alta prioridade aqui

Quadro DCF regular aqui

Quadro de baixa prioridade aqui

Recuperação de quadro ruim feita

Tempo

Figura 4-28. Espaçamento entre quadros em 802.11.

Os dois intervalos AIFS (**A**rbitration **I**nterFrame **S**pace) mostram exemplos de dois níveis de prioridade diferentes. O curto intervalo, AIFS 1 , é menor do que DIFS, mas mais do que SIFS. Pode ser usado pelo AP para mover voz ou outra alta prioridade

SEC. 4,4

LANS SEM FIO

309

tráfego para o início da linha. O AP irá esperar por um intervalo menor antes de

envia o tráfego de voz e, portanto, o envia antes do tráfego normal. O longo intervalo, AIFS 4 , é maior do que DIFS. É usado para tráfego de fundo que pode ser adiado até depois do tráfego regular. O AP irá esperar por um intervalo maior antes de enviar este tráfego, dando ao tráfego regular a oportunidade de transmitir primeiro. O completo mecanismo de qualidade de serviço define quatro níveis de prioridade diferentes que têm diferentes diferentes parâmetros de backoff, bem como diferentes parâmetros de inatividade.

O último intervalo de tempo, **EIFS** (**Extended InterFrame Spacing**), é usado apenas por uma estação que acabou de receber um quadro inválido ou desconhecido, para relatar o problema.

A ideia é que, uma vez que o receptor pode não ter ideia do que está acontecendo, ele deve espere um pouco para evitar interferir em um diálogo em andamento entre duas estações.

Outra parte da qualidade das extensões de serviço é a noção de um **TXOP** ou **oportunidade de transmissão** . O mecanismo CSMA / CA original permite que as estações enviem um quadro de cada vez. Este projeto funcionou bem até que a faixa de taxas aumentou. Com 802.11a / g, uma estação pode estar enviando a 6 Mbps e outra estação pode estar enviando a 54 Mbps. Cada um deles envia um quadro, mas a estação de 6 Mbps leva nove tempos tão longos (ignorando sobrecargas fixas) quanto a estação de 54 Mbps para enviar seu quadro.

Essa disparidade tem o infeliz efeito colateral de desacelerar um remetente rápido que está competindo com um remetente lento para aproximadamente a taxa do remetente lento. Por exemplo, novamente ignorando sobrecargas fixas, ao enviar sozinho os 6 Mbps e 54 Mbps os remetentes receberão suas próprias taxas, mas ao enviarem juntos, ambos receberão 5,4 Mbps em média. É uma penalidade rígida para o remetente rápido. Este problema é conhecido como a **anomalia de taxa** (Heusse et al., 2003).

Com oportunidades de transmissão, cada estação recebe uma quantidade igual de tempo de antena, não um número igual de frames. Estações que enviam a uma taxa mais alta pelo tempo de antena obterá maior rendimento. Em nosso exemplo, ao enviar juntos os 6 Mbps e Os remetentes de 54 Mbps terão agora 3 Mbps e 27 Mbps, respectivamente.

4.4.4 A Estrutura do Frame 802.11

O padrão 802.11 define três classes diferentes de frames no ar: dados, controle e gestão. Cada um deles tem um cabeçalho com uma variedade de campos usados dentro da subcamada MAC. Além disso, existem alguns cabeçalhos usados pelo físico camada cal, mas estes lidam principalmente com as técnicas de modulação usadas, então vamos não os discutir aqui.

Veremos o formato do quadro de dados como exemplo. É mostrado em Fig. 4-29. Primeiro, vem o campo de *controle Frame* , que é composto de 11 subcampos. A primeira delas é a *versão do protocolo* , definida como 00. Ela existe para permitir *versões* futuras sões de 802.11 para operar ao mesmo tempo na mesma célula. Então vem o *tipo* (dados, controle ou gerenciamento) e campos de *subtipo* (por exemplo, RTS ou CTS). Para um regulamento quadro de dados lar (sem qualidade de serviço), eles são definidos como 10 e 0000 em binário. Os bits *To DS* e *From DS* são definidos para indicar se o quadro está indo para ou vindo da rede conectada aos APs, que é chamada de distribuição

310

O SUBCAMADA DE CONTROLE DE ACESSO MÉDIO INDIVÍDUO. 4

sistema. O bit *Mais fragmentos* significa que mais fragmentos virão. o O bit de nova *tentativa* marca uma retransmissão de um quadro enviado anteriormente. O *gerenciamento de energia*

bit indica que o remetente está entrando no modo de economia de energia. O bit *Mais de dados* em indica que o remetente tem quadros adicionais para o receptor. O *protegido*

O bit do *quadro* indica que o corpo do quadro foi criptografado para segurança. Nós vamos discutir a segurança brevemente na próxima seção. Finalmente, o bit de *pedido* informa ao receptor que a camada superior espera que a sequência de quadros chegue estritamente em ordem.

2
2
2
0-2312
Seqüência
Endereço 1
(destinatário)
Duração
Dados
Quadro, Armação
ao controle
Verifica
seqüência
4
6
6
6
Endereço 2
(transmissor)
Endereço 3
2
2
1
1
Subtipo
= 0000
Tipo
= 10
Versão
= 00
4
1
Para
DS
De
DS
Mais
frag.
Tentar novamente
Pwr.
mgt.
Mais
dados
Ordem Protegida
1
1
1
1
1
Bits

Figura 4-29. Formato do quadro de dados 802.11.

O segundo campo do quadro de dados, o campo *Duração*, informa por quanto tempo o quadro e seu reconhecimento ocupará o canal, medido em microsegundos onds. Está presente em todos os tipos de frames, incluindo frames de controle, e é o que estações usam para gerenciar o mecanismo NAV .

Em seguida, vêm os endereços. Os frames de dados enviados para ou de um AP têm três ad- vestidos, todos no formato padrão IEEE 802. O primeiro endereço é o receptor, e o o segundo endereço é o transmissor. Eles são obviamente necessários, mas qual é o terceiro endereço para? Lembre-se de que o AP é simplesmente um ponto de retransmissão para quadros enquanto eles

viajar entre um cliente e outro ponto na rede, talvez um cliente distante ou um portal para a Internet. O terceiro endereço fornece este ponto final distante. O campo *Sequência* numera quadros para que as duplicatas possam ser detectadas. Do 16 bits disponíveis, 4 identificam o fragmento e 12 carregam um número avançado com cada nova transmissão. O campo de *dados* contém a carga útil, até 2312 bytes. Os primeiros bytes desta carga útil estão em um formato conhecido como **LLC (lógico Controle de link)**. Esta camada é a cola que identifica o protocolo da camada superior (por exemplo, IP) para o qual as cargas úteis devem ser passadas. Por último vem a *verificação do quadro*

sequência, que é o mesmo de 32 bits CRC vimos no cap. 3.2.2 e em outros lugares.

Os quadros de gerenciamento têm o mesmo formato dos quadros de dados, mais um formato para a parte de dados que varia com o subtipo (por exemplo, parâmetros em quadros de beacon).

Os quadros de controle são curtos. Como todos os quadros, eles têm o *controle de quadro*, *duração*, e campos de *sequência de verificação de quadro*. No entanto, eles podem ter apenas um endereço e

nenhuma porção de dados. A maioria das informações principais é transmitida com o campo *Subtipo* (por exemplo, ACK , RTS e CTS).

Página 335

SEC. 4,4
LANS SEM FIO

311

4.4.5 Serviços

O padrão 802.11 define os serviços que os clientes, os pontos de acesso, e a rede que os conecta deve ser uma LAN sem fio em conformidade. Estes serviços se agrupam em vários grupos.

O serviço de **associação** é usado por estações móveis para se conectar a APs. Normalmente, ele é usado logo depois que uma estação se move dentro do alcance de rádio do AP.

Após a chegada, a estação aprende a identidade e as capacidades do AP, seja de beacon frames ou perguntando diretamente ao AP. Os recursos incluem as taxas de dados com suporte, disposições de segurança, recursos de economia de energia, qualidade de serviço suporte e muito mais. A estação envia uma solicitação para se associar ao AP. O AP pode aceitar ou rejeitar o pedido.

A **reassociação** permite que uma estação mude seu AP preferencial. Esta facilidade é útil para estações móveis movendo-se de um AP para outro AP na mesma extensão LAN 802.11, como um handover na rede celular. Se for usado corretamente, não os dados serão perdidos como consequência da transferência. (Mas 802.11, como Ethernet, é apenas um serviço de melhor esforço.) A estação ou o AP também podem se **desassociar**, quebrando seu relacionamento. Uma estação deve usar este serviço antes de desligar ou saindo da rede. O AP pode usá-lo antes de ir para manutenção.

As estações também devem se **autenticar** antes de enviarem frames através do AP, mas a autenticação é tratada de maneiras diferentes, dependendo da escolha de segurança esquema. Se a rede 802.11 estiver "aberta", qualquer pessoa pode usá-la. De outra forma, são necessárias credenciais para autenticar. O esquema recomendado, chamado **WPA2** (**WiFi Protected Access 2**), implementa a segurança conforme definido na norma 802.11i dard. (WPA simples é um esquema provisório que implementa um subconjunto de 802.11i. irá pular e ir direto para o esquema completo.) Com WPA2, o AP pode falar para um servidor de autenticação que tem um banco de dados de nome de usuário e senha para determinar

meu se a estação tem permissão para acessar a rede. Alternativamente, um pré-compartilhado chave, que é um nome fantasia para uma senha de rede, pode ser configurada. De várias quadros são trocados entre a estação e o AP com um desafio e re- que permite à estação provar que tem as credenciais certas. Esta troca aconteceu canetas após associação.

O esquema que foi usado antes do WPA é chamado **WEP** (**Wired Equivalent Privacidade**). Para este esquema, a autenticação com uma chave pré-compartilhada acontece antes Associação. No entanto, seu uso é desencorajado por causa de falhas de design que tornam WEP fácil de comprometer. A primeira demonstração prática de que WEP era bro- ken veio quando Adam Stubblefield era um estagiário de verão na AT&T (Stubblefield et al., 2002). Ele foi capaz de codificar e testar um ataque em uma semana, muitos dos quais foi gasto para obter permissão da administração para comprar os cartões WiFi necessários para experimentos. Software para quebrar senhas WEP agora está disponível gratuitamente.

Uma vez que os quadros chegam ao AP, o serviço de **distribuição** determina como rotear eles. Se o destino for local para o AP, os quadros podem ser enviados diretamente por o ar. Caso contrário, eles terão que ser encaminhados pela rede com fio. o

Página 336

312

O SUBCAMADA DE CONTROLE DE ACESSO MÉDIO
INDIVÍDUO. 4

serviço de **integração** lida com qualquer tradução necessária para que um quadro seja enviado fora da LAN 802.11 ou para chegar de fora da LAN 802.11. O comum caso aqui é conectar a LAN sem fio à Internet.

A transmissão de dados é o que importa, então 802.11 naturalmente fornece **dados** serviço de **entrega**. Este serviço permite que as estações transmitam e recebam dados usando os protocolos que descrevemos anteriormente neste capítulo. Uma vez que 802.11 é modelado em Ether-

rede e transmissão através de Ethernet não têm garantia de ser 100% confiável, transmissão sobre 802.11 também não é garantida como confiável. Camadas mais altas devem lidar com a detecção e correção de erros.

Wireless é um sinal de transmissão. Para informações enviadas através de uma LAN sem fio para ser mantido em sigilo, deve ser criptografado. Este objetivo é alcançado com um **pri**-serviço **vacy** que gerencia os detalhes de criptografia e descriptografia. A criptografia algoritmo de operação para WPA2 é baseado em **AES** (**Advanced Encryption Standard**), um Padrão do governo dos EUA aprovado em 2002. As chaves usadas para en- a criptografia é determinada durante o procedimento de autenticação.

Para lidar com o tráfego com prioridades diferentes, há um **agendamento de tráfego QOS** serviço. Ele usa os protocolos que descrevemos para fornecer tráfego de voz e vídeo antes tratamento diferencial comparado ao melhor esforço e tráfego de fundo. Uma companhia serviço também fornece sincronização de temporizador de camada superior. Isso permite que as estações coordenem

nate suas ações, o que pode ser útil para o processamento de mídia.

Por fim, existem dois serviços que ajudam as estações a gerenciar o uso da especificação trum. O serviço de **controle de potência de transmissão** fornece às estações as informações que elas

precisam atender aos limites regulamentares de potência de transmissão que variam de região para região.

O serviço de **seleção dinâmica de frequência** fornece às estações as informações de que precisam para evitar a transmissão em frequências na banda de 5 GHz que estão sendo usadas para radar nas proximidades.

Com esses serviços, 802.11 fornece um rico conjunto de funcionalidades para conectar clientes móveis próximos à Internet. Foi um grande sucesso, e o padrão foi alterado repetidamente para adicionar mais funcionalidade. Para uma perspectiva sobre onde o padrão esteve e para onde está indo, consulte Hiertz et al. (2010).

4.5 BANDA LARGA WIRELESS

Ficamos dentro de casa por muito tempo. Vamos lá fora, onde há bastante de redes interessantes na chamada "última milha". Com a desregulamentação de os sistemas de telefonia em muitos países, concorrentes do telefone entrincheirado as empresas agora têm permissão para oferecer voz local e serviços de Internet de alta velocidade vice. Certamente há muita demanda. O problema é que correr fibra ou coaxial para milhões de lares e empresas é proibitivamente caro. O que é uma concorrente a fazer?

A resposta é banda larga sem fio. Erguer uma grande antena em uma colina logo ali - lado da cidade é muito mais fácil e barato do que cavar muitas trincheiras e amarrar

cabos. Assim, as empresas começaram a experimentar o fornecimento de multimegabit serviços de comunicação sem fio para voz, Internet, filmes sob demanda, etc.

Para estimular o mercado, o IEEE formou um grupo para padronizar uma banda larga rede de área metropolitana sem fio. O próximo número disponível no número 802 o espaço da bering era **802,16**, então o padrão obteve esse número. Informalmente a tecnologia ogy é chamado de **WiMAX** (**Worldwide Interoperability for Microwave Access**).

Usaremos os termos 802.16 e WiMAX alternadamente.

O primeiro padrão 802.16 foi aprovado em dezembro de 2001. Versões anteriores

forneceu um loop local sem fio entre pontos fixos com uma linha de visão para cada de outros. Este design logo mudou para tornar o WiMAX uma alternativa mais competitiva para cabo e DSL para acesso à Internet. Em janeiro de 2003, 802.16 foi revisado para oferecer suporte a links sem linha de visão usando a tecnologia OFDM em frequências entre 2 GHz e 10 GHz. Essa mudança tornou a implantação muito mais fácil, embora as estações ainda eram locais fixos. A ascensão das redes celulares 3G representa uma ameaça prometendo altas taxas de dados e mobilidade. Em resposta, 802.16 foi aprimorado novamente para permitir a mobilidade em velocidades veiculares até dezembro de 2005. o acesso de banda à Internet é o alvo do padrão atual, IEEE 802.16-2009.

Como os outros padrões 802, 802.16 foi fortemente influenciado pelo OSI modelo, incluindo as (sub) camadas, terminologia, primitivas de serviço e muito mais. Un-felizmente, também como o OSI, é bastante complicado. Na verdade, o **WiMAX Forum** foi criado para definir subconjuntos interoperáveis do padrão para oferta comercial ings. Nas seções a seguir, daremos uma breve descrição de alguns dos destaques das formas comuns de interface aérea 802.16, mas este tratamento está longe de completo e omite muitos detalhes. Para obter informações adicionais sobre WiMAX e banda larga sem fio em geral, consulte Andrews et al. (2007).

4.5.1 Comparação de 802.16 com 802.11 e 3G

Neste ponto, você pode estar pensando: por que criar um novo padrão? Porque não apenas usa 802.11 ou 3G? Na verdade, o WiMAX combina aspectos de 802.11 e 3G, tornando-o mais parecido com uma tecnologia 4G.

Assim como o 802.11, o WiMAX tem tudo a ver com conectar dispositivos sem fio ao Inter net em velocidades megabit / seg, em vez de usar cabo ou DSL. Os dispositivos podem ser móvel, ou pelo menos portátil. WiMAX não começou adicionando dados de baixa taxa no lado de redes celulares semelhantes a voz; 802.16 foi projetado para transportar pacotes IP sobre o ar e para se conectar a uma rede com fio baseada em IP com o mínimo de confusão. o os pacotes podem transportar tráfego ponto a ponto, chamadas VoIP ou mídia de streaming para suportar um

gama de aplicações. Também como 802.11, é baseado na tecnologia OFDM para garantir um bom desempenho, apesar da degradação do sinal sem fio, como desvanecimento tipath e na tecnologia MIMO para atingir altos níveis de rendimento.

No entanto, o WiMAX é mais parecido com 3G (e, portanto, diferente de 802.11) em vários respects. O principal problema técnico é alcançar alta capacidade pelo uso eficiente do espetro, de modo que um grande número de assinantes em uma área de cobertura possa obter

Página 338

314

O SUBCAMADA DE CONTROLE DE ACESSO MÉDIO INDIVÍDUO. 4

alto rendimento. As distâncias típicas são pelo menos 10 vezes maiores do que para um Rede 802.11. Consequentemente, as estações base WiMAX são mais poderosas do que Pontos de acesso 802.11 (APs). Para lidar com sinais mais fracos em distâncias maiores, o estação base usa mais energia e melhores antenas, e executa mais proc- essing para lidar com erros. Para maximizar o rendimento, as transmissões são cuidadosamente programado pela estação base para cada assinante particular; o uso do espetro não é deixado ao acaso com o CSMA / CA, que pode desperdiçar capacidade com as colisões. O espetro licenciado é o caso esperado para WiMAX, normalmente em torno de 2,5 GHz nos EUA. Todo o sistema é substancialmente mais otimizado do que 802.11.

Essa complexidade vale a pena, considerando a grande quantidade de dinheiro envolvida para espetro licenciado. Ao contrário do 802.11, o resultado é um serviço gerenciado e confiável com bom suporte para qualidade de serviço.

Com todos esses recursos, o 802.16 mais se assemelha à rede celular 4G trabalhos que agora estão sendo padronizados sob o nome de **LTE (Long Term Evolu- ção)**. Enquanto as redes celulares 3G são baseadas em CDMA e suportam voz e de dados, as redes de celular 4G serão baseadas em OFDM com MIMO, e terão tar- obter dados, com a voz como um único aplicativo. Parece que WiMAX e 4G estão em um curso de colisão em termos de tecnologia e aplicações. Talvez este conver-

gente não é surpreendente, dado que a Internet é o aplicativo matador e OFDM e MIMO são as tecnologias mais conhecidas para o uso eficiente do espectro.

4.5.2 A arquitetura 802.16 e pilha de protocolo

A arquitetura 802.16 é mostrada na Figura 4-30. As estações base se conectam diretamente à rede de backbone do provedor, que por sua vez está conectado à Internet.

As estações base se comunicam com as estações por meio da interface aérea sem fio. Dois tipos de estações existem. As estações de assinantes permanecem em um local fixo, por exemplo, acesso à Internet de banda larga para residências. Estações móveis podem receber serviço enquanto eles estão movendo, por exemplo, um carro equipado com WiMAX.

A pilha de protocolo 802.16 que é usada na interface aérea é mostrada em Fig. 4-31. A estrutura geral é semelhante às das outras redes 802, mas com mais subcamadas. A camada inferior trata da transmissão, e aqui temos mostrado apenas as ofertas populares de 802.16, WiMAX fixo e móvel. Há sim uma camada física diferente para cada oferta. Ambas as camadas operam em especificações licenciadas

trum abaixo de 11 GHz e usar OFDM, mas de maneiras diferentes.

Acima da camada física, a camada de enlace de dados consiste em três subcamadas. O último lida com privacidade e segurança, que é muito mais crucial para o público redes externas do que para redes privadas internas. Ele gerencia a criptografia, de-cryptografia e gerenciamento de chaves.

Em seguida, vem a parte da subcamada comum do MAC. Esta parte é onde os principais protocolos, como gerenciamento de canal, estão localizados. O modelo aqui é que a estação base controla completamente o sistema. Ele pode agendar o downlink (ou seja, de base para assinantes) canais de forma muito eficiente e desempenha um papel importante na gestão

Página 339

SEC. 4,5
BROADBAND WIRELESS

315

Base
estação
móvel
estações
Assinante
estações
Rede de backbone
(para Internet)

Interface aérea

Figura 4-30. A arquitetura 802.16.

“WiMAX fixo”
OFDM (802.16a)
“Mobile WiMAX”
OFDMA escalável (802.16e)

Subcamada de convergência específica de serviço
Data de lançamento:

2003
2005

Superior
camadas
Link de dados
camada

Física

camada

Subcamada comum MAC

Subcamada de segurança

IP, por exemplo

Figura 4-31. A pilha do protocolo 802.16.

os canais de uplink (ou seja, assinante para base) também. Uma característica incomum deste A subcamada MAC é que, ao contrário dos outros protocolos 802, é completamente orientada para conexão, a fim de fornecer garantias de qualidade de serviço para telecomunicação falsa e multimídia.

A subcamada de convergência específica do serviço toma o lugar do link lógico subcamada nos outros protocolos 802. Sua função é fornecer uma interface para o camada de rede. Diferentes camadas de convergência são definidas para se integrarem perfeitamente com diferentes camadas superiores. A escolha importante é o IP, embora o padrão

define mapeamentos para protocolos como Ethernet e ATM também. Uma vez que IP é conexão e a subcamada 802.16 MAC é orientada para conexão, esta camada deve mapear entre endereços e conexões.

316

O SUBCAMADA DE CONTROLE DE ACESSO MÉDIO
INDIVÍDUO. 4

4.5.3 A Camada Física 802.16

A maioria das implantações de WiMAX usa espectro licenciado em torno de 3,5 GHz ou 2,5 GHz. Tal como acontece com o 3G, encontrar o espectro disponível é um problema chave. Para ajudar, o

O padrão 802.16 é projetado para flexibilidade. Permite operação de 2 GHz a 11 GHz. Canais de tamanhos diferentes são suportados, por exemplo, 3,5 MHz para WiMAX e de 1,25 MHz a 20 MHz para WiMAX móvel.

As transmissões são enviadas por meio desses canais com OFDM, a técnica que desenhamos riscado na Seç. 2.5.3. Comparado com 802.11, o design 802.16 OFDM é otimizado para tirar o máximo proveito do espectro licenciado e das transmissões de área ampla. O canal é dividido em mais subportadoras com uma duração de símbolo mais longa para tolerar maiores degradações do sinal wireless; Os parâmetros de WiMAX são cerca de 20 vezes maiores mais do que parâmetros 802.11 comparáveis. Por exemplo, em WiMAX móvel, existem 512 subportadoras para um canal de 5 MHz e o tempo para enviar um símbolo em cada a subportadora tem aproximadamente 100 µseg.

Os símbolos em cada subportadora são enviados com QPSK, QAM-16 ou QAM-64, mod-esquemas de utilização que descrevemos na Seç. 2.5.3. Quando o celular ou assinante estiver está perto da estação base e o sinal recebido tem uma alta relação sinal-ruído (SNR), QAM-64 pode ser usado para enviar 6 bits por símbolo. Para alcançar estações distantes com um SNR baixo, QPSK pode ser usado para fornecer 2 bits por símbolo. Os dados são os primeiros

codificado para correção de erros com a codificação convolucional (ou melhores esquemas) que que descrevemos na Seç. 3.2.1. Esta codificação é comum em canais barulhentos para tolerar alguns erros de bit sem a necessidade de enviar retransmissões. Na verdade, a modulação e os métodos de codificação devem soar familiares agora, pois são usados para muitos trabalhos que estudamos, incluindo cabo 802.11 e DSL. O resultado líquido é que um a estação base pode suportar até 12,6 Mbps de tráfego de downlink e 6,2 Mbps de tráfego de uplink por canal de 5 MHz e par de antenas.

Uma coisa que os designers do 802.16 não gostaram foi um certo aspecto da maneira GSM e DAMPS funcionam. Ambos os sistemas usam bandas de frequência iguais para tráfego upstream e downstream. Ou seja, eles assumem implicitamente que há tanto tráfego upstream como tráfego downstream. Para voz, o tráfego é simétrico para o na maior parte, mas para acesso à Internet (e certamente navegação na Web), muitas vezes há mais tráfego downstream do que tráfego upstream. A proporção geralmente é 2: 1, 3: 1 ou mais: 1. Então, os designers escolheram um esquema flexível para dividir o canal entre

estações, chamadas **OFDMA (Orthogonal Frequency Division Multiple Access)**.

Com OFDMA, diferentes conjuntos de subportadoras podem ser atribuídos a diferentes estações, para que mais de uma estação possa enviar ou receber ao mesmo tempo. Se fosse 802.11, todos subportadoras seriam usadas por uma estação para enviar a qualquer momento. O adicionado flexibilidade em como a largura de banda é atribuída pode aumentar o desempenho porque um determinada subportadora pode estar desbotada em um receptor devido aos efeitos de multipercorso, mas claro

em outro. As subportadoras podem ser atribuídas às estações que podem usá-las melhor.

Além de ter tráfego assimétrico, as estações geralmente alternam entre enviar receber e receber. Este método é denominado **TDD (Time Division Duplex)**. o

método alternativo, em que uma estação envia e recebe ao mesmo tempo (em diferentes frequências de subportadora), é denominado **FDD** (**Frequency Division Duplex**). O WiMAX permite os dois métodos, mas o TDD é o preferido porque é mais fácil de implementar e mais flexível.

```

Guarda
Ranging
Estouro
Estouro
Estouro
Estouro
Estouro
Estouro
Estouro
Estouro
Estouro
Downlink
mapa
Uplink
mapa
Preâmbulo
Tempo
Subportadora
Uplink
Downlink
Próximo
quadro, Armação
Último
quadro, Armação

```

Figura 4-32. Estrutura de quadro para OFDMA com duplexação por divisão de tempo.

A Figura 4-32 mostra um exemplo da estrutura do quadro que se repete ao longo do tempo. Ele começa com um preâmbulo para sincronizar todas as estações, seguido por downlink transmissões da estação base. Primeiro, a estação base envia mapas que informam a todas as estações informações sobre como as subportadoras de downlink e uplink são atribuídas no quadro. O a estação base controla os mapas, para que possa alocar diferentes quantidades de largura de banda às estações quadro a quadro dependendo das necessidades de cada estação.

Em seguida, a estação base envia rajadas de tráfego para diferentes assinantes e dispositivos móveis estações nas subportadoras nos horários indicados no mapa. A transmissão do downlink As sessões terminam com um tempo de guarda para que as estações mudem de recepção para transmissão.

Finalmente, o assinante e as estações móveis enviam seus surtos de tráfego para a base estação nas posições de uplink que foram reservadas para eles no mapa. Um de esses bursts de uplink são reservados para **variação**, que é o processo pelo qual novas estações ajustam seu tempo e solicitam largura de banda inicial para se conectar à base estação. Uma vez que nenhuma conexão é configurada nesta fase, novas estações apenas transmitem e espero que não haja colisão.

4.5.4 O protocolo 802.16 MAC Sublayer

A camada de enlace de dados é dividida em três subcamadas, como vimos na Figura 4.31.

Uma vez que não estudaremos criptografia até o cap. 8, é difícil explicar agora como funciona a subcamada de segurança. Basta dizer que a criptografia é usada para manter segredo todos os dados transmitidos. Apenas os payloads do quadro são criptografados; os cabeçalhos

não são. Esta propriedade significa que um bisbilhoteiro pode ver quem está falando com quem, mas não podem dizer o que estão dizendo um ao outro.

Se você já sabe algo sobre criptografia, o que se segue é um explicação do parágrafo da subcamada de segurança. Se você não sabe nada sobre criptograficamente, não é provável que você ache o próximo parágrafo terrivelmente esclarecedor (mas você pode considerar relê-lo depois de terminar o cap. 8).

Quando um assinante se conecta a uma estação base, eles realizam autenticação mútua com criptografia de chave pública RSA usando certificados X.509. As cargas úteis são criptografados usando um sistema de chave simétrica, AES (Rijndael) ou DES com encadeamento de blocos de cifras. A verificação de integridade usa SHA-1. Agora isso não foi tão ruim, foi?

Vejamos agora a parte da subcamada comum do MAC. A subcamada MAC é orientado para conexão e ponto-a-multiponto, o que significa que uma estação base se comunica com várias estações de assinantes. Muito deste design é emprestado de modems a cabo, em que um headend de cabo controla as transmissões de vários modems a cabo nas instalações do cliente.

A direção do downlink é bastante direta. A estação base controla o rajadas de camada física que são usadas para enviar informações para diferentes assinantes estações. A subcamada MAC simplesmente compacta seus quadros nessa estrutura. Reduzir sobrecarga, existem várias opções diferentes. Por exemplo, frames MAC podem ser enviados individualmente ou embalados consecutivamente em um grupo.

O canal de uplink é mais complicado, pois há inscritos concorrentes que precisam de acesso a ele. Sua alocação está intimamente ligada à questão da qualidade do serviço.

Quatro classes de serviço são definidas, como segue:

1. Serviço de taxa de bits constante.
2. Serviço de taxa de bits variável em tempo real.
3. Serviço de taxa de bits variável não em tempo real.
4. Serviço de melhor esforço.

Todos os serviços em 802.16 são orientados à conexão. Cada conexão obtém um destes classes de serviço, determinadas quando a conexão é configurada. Este design é diferente daquele de 802.11 ou Ethernet, que não têm conexão na subcamada MAC.

O serviço de taxa de bits constante se destina à transmissão de voz não compactada.

Este serviço precisa enviar uma quantidade predeterminada de dados em um tempo predeterminado intervalos. É acomodado dedicando certas rajadas para cada conexão de esse tipo. Uma vez que a largura de banda foi alocada, os bursts estão disponíveis automaticamente maticamente, sem a necessidade de pedir para cada um.

O serviço de taxa de bits variável em tempo real é para multimídia compactada e outros aplicativos soft em tempo real em que a quantidade de largura de banda necessária em cada stant pode variar. Ele é acomodado pela estação base que consulta o assinante em um intervalo fixo para perguntar quanta largura de banda é necessária neste momento.

Página 343

SEC. 4,5
BROADBAND WIRELESS

319

O serviço de taxa de bits variável não em tempo real é para transmissões pesadas que não são em tempo real, como transferências de arquivos grandes. Para este serviço, a estação base consulta o assinante frequentemente, mas não em intervalos de tempo rigidamente prescritos. Conexões com este serviço também pode usar o serviço de melhor esforço, descrito a seguir, para solicitar largura de banda.

O serviço de melhor esforço é para todo o resto. Nenhuma votação é feita e o sub O escriba deve disputar largura de banda com outros assinantes de melhor esforço. solicitações de para largura de banda são enviados em rajadas marcadas no mapa de uplink como disponíveis para con- atenção. Se uma solicitação for bem-sucedida, seu sucesso será observado no próximo downlink mapa. Se não for bem-sucedido, o assinante malsucedido deverá tentar novamente mais tarde. Para Para minimizar as colisões, o algoritmo de backoff exponencial binário Ethernet é usado.

4.5.5 A Estrutura do Quadro 802.16

Todos os quadros MAC começam com um cabeçalho genérico. O cabeçalho é seguido por um carga útil opcional e uma soma de verificação opcional (CRC), conforme ilustrado na Fig. 4-33. A carga útil não é necessária em quadros de controle, por exemplo, aqueles que solicitam alteração

slots nel. A soma de verificação (surpreendentemente) também é opcional, devido à correção de erros

na camada física e o fato de que nenhuma tentativa é feita para retransmitir prazos. Se nenhuma retransmissão será tentada, por que se preocupar com um soma de verificação? Mas se houver uma soma de verificação, é o padrão IEEE 802 CRC e conhecimentos e retransmissões são usados para confiabilidade.

Bits
(uma)
(b)
Tipo
comprimento
0
1 0
Tipo
Bytes necessários
EK
E
C
C
Eu
ID de conexão
Dados
CRC
Cabeçalho
CRC
ID de conexão
Cabeçalho
CRC
1 1
6
16
16
8
1 1
1 1 12
6
11
16
8
4
Bits

Figura 4-33. (a) Um quadro genérico. (b) Um quadro de solicitação de largura de banda.

Segue um rápido resumo dos campos de cabeçalho da Figura 4.33 (a). O bit *EC* diz se a carga útil é criptografada. O campo *Tipo* identifica o tipo de quadro, principalmente informando se o empacotamento e a fragmentação estão presentes. O campo *CI* indica a presença ou ausência da soma de verificação final. O campo *EK* diz qual as chaves de criptografia estão sendo usadas (se houver). O campo *Comprimento* fornece o completo comprimento do quadro, incluindo o cabeçalho. O *identificador de conexão* diz qual conexão a que este quadro pertence. Finalmente, o campo *Header CRC* é uma soma de verificação sobre o cabeçalho apenas, usando o polinômio $x^8 + x^2 + x + 1$.

O protocolo 802.16 possui muitos tipos de quadros. Um exemplo de um diferente tipo de quadro usado para solicitar largura de banda é mostrado na Figura 4.33 (b). isto

320

O SUBCAMADA DE CONTROLE DE ACESSO MÉDIO INDIVÍDUO. 4

começa com 1 bit em vez de 0 e é semelhante ao cabeçalho genérico exceto que o segundo e o terceiro bytes formam um número de 16 bits informando quanto a largura de banda é necessária para transportar o número especificado de bytes. Pedido de largura de banda

os quadros não carregam uma carga útil ou CRC de quadro completo.

Muito mais poderia ser dito sobre 802.16, mas este não é o lugar para dizer isto. Para obter mais informações, consulte o próprio padrão IEEE 802.16-2009.

4.6 BLUETOOTH

Em 1994, a empresa LM Ericsson ficou interessada em conectar seus telefones celulares para outros dispositivos (por exemplo, laptops) sem cabos. Junto com quatro outras empresas (IBM, Intel, Nokia e Toshiba), formou um SIG (Special Interest Group, ou seja, consórcio) em 1998 para desenvolver um padrão sem fio para interconexão

conectar dispositivos e acessórios de computação e comunicação usando curto alcance, rádios sem fio baratos e de baixo consumo. O projeto foi denominado **Bluetooth**, após Harald Blaatand (Bluetooth) II (940-981), um rei Viking que unificou (ou seja, questionado) Dinamarca e Noruega, também sem cabos.

O Bluetooth 1.0 foi lançado em julho de 1999 e, desde então, o SIG nunca olhou para trás. Todos os tipos de dispositivos eletrônicos de consumo agora usam Bluetooth, de telefones celulares e laptops para fones de ouvido, impressoras, teclados, mouses, gameboxes, relógios, reprodutores de música, unidades de navegação e muito mais. Os protocolos Bluetooth permitem que esses dispositivos se encontram e se conectam, um ato chamado **emparelhamento**, e com segurança transferir dados.

Os protocolos também evoluíram na última década. Após o proto inicial colo estabilizou, taxas de dados mais altas foram adicionadas ao Bluetooth 2.0 em 2004. Com o Versão 3.0 em 2009, o Bluetooth pode ser usado para emparelhamento de dispositivos em combinação com 802.11 para transferência de dados de alto rendimento. A versão 4.0 em dezembro de 2009 especifica operação de baixa potência instalada. Isso será útil para pessoas que não querem trocar as baterias regularmente em todos os dispositivos da casa. Nós vamos cobrir os principais aspectos do Bluetooth abaixo.

4.6.1 Arquitetura Bluetooth

Vamos começar nosso estudo do sistema Bluetooth com uma rápida visão geral do que ele contém e o que se destina a fazer. A unidade básica de um sistema Bluetooth é um **piconet**, que consiste em um nó mestre e até sete nós escravos ativos com uma distância de 10 metros. Podem existir várias piconets na mesma sala (grande) e pode até ser conectado por meio de um nó de ponte que participa de várias piconets, como na Fig. 4-34. Uma coleção interconectada de piconets é chamada de **scatternet**. Além dos sete nós escravos ativos em uma piconet, pode haver até 255 nós estacionados na rede. Estes são dispositivos que o mestre mudou para um baixo estado de energia para reduzir o consumo de suas baterias. No estado estacionado, um dispositivo não pode

Página 345

SEC. 4,6
BLUETOOTH

321

S
S
S
S
S
S
S
S
S
S
M
M
Escravo da ponte
Estacionado
escravo
Piconet 2
Piconet 1
Ativo
escravo

Figura 4-34. Duas piconets podem ser conectadas para formar uma scatternet.
faça qualquer coisa, exceto responder a um sinal de ativação ou beacon do mestre.
Dois estados de potência intermediários, hold e sniff, também existem, mas estes não aparecem-nos aqui.

A razão para o projeto mestre / escravo é que os projetistas pretendiam facilitar a implementação de chips Bluetooth completos por menos de \$ 5. A consequência desta decisão é que os escravos são bastante burros, basicamente apenas fazendo tudo o que o mestre lhes disser para fazer. Em sua essência, uma piconet é um TDM centralizado

sistema, com o mestre controlando o relógio e determinando qual dispositivo obtém para se comunicar em qual intervalo de tempo. Toda a comunicação é entre o mestre e um escravo; a comunicação escravo-escravo direta não é possível.

4.6.2 Aplicativos Bluetooth

A maioria dos protocolos de rede apenas fornece canais entre entidades de comunicação vincula e permite que os designers de aplicativos descubram para que querem usá-los. Para exemplo, 802.11 não especifica se os usuários devem usar seus notebooks computadores para ler e-mail, navegar na Web ou qualquer outra coisa. Em contraste, o Bluetooth SIG especifica aplicativos específicos a serem suportados e oferece diferentes pilhas de protocolo diferentes para cada um. No momento em que este artigo foi escrito, havia 25 aplicativos

, que são chamados de **perfis**. Infelizmente, essa abordagem leva a um grande quantidade de complexidade. Omitiremos a complexidade aqui, mas examinaremos brevemente os perfis para ver mais claramente o que o Bluetooth SIG está tentando realizar.

Seis dos perfis são para diferentes usos de áudio e vídeo. Por exemplo, o perfil de intercomunicação permite que dois telefones se conectem como walkie-talkies. O fone de ouvido e perfis de mãos-livres fornecem comunicação de voz entre um fone de ouvido e sua estação base, que pode ser usada para telefonia viva-voz ao dirigir um carro.

Página 346

322

O SUBCAMADA DE CONTROLE DE ACESSO MÉDIO INDIVÍDUO. 4

Outros perfis são para streaming de áudio e vídeo com qualidade estéreo, digamos, de uma porta reprodutor de música capaz para fones de ouvido ou de uma câmera digital para uma TV.

O perfil do dispositivo de interface humana serve para conectar teclados e mouses a computadores. Outros perfis permitem que um telefone celular ou outro computador receba imagens de uma câmera ou envie imagens para uma impressora. Talvez seja de mais interesse um perfil para usar um telefone celular como controle remoto para uma TV (habilitada para Bluetooth).

Ainda outros perfis permitem a rede. O perfil de rede de área pessoal permite Dispositivos Bluetooth formam uma rede ad hoc ou acessam remotamente outra rede, como uma LAN 802.11, por meio de um ponto de acesso. O perfil de rede dial-up era na verdade, a motivação original para todo o projeto. Ele permite que um notebook comp computador para se conectar a um telefone celular contendo um modem embutido sem usar fios.

Perfis para troca de informações de camadas superiores também foram definidos. o perfil de sincronização destina-se a carregar dados em um telefone celular quando sai de casa e coleta dados quando ele retorna.

Pularemos o resto dos perfis, exceto para mencionar que alguns perfis servem como blocos de construção sobre os quais os perfis acima são construídos. O acesso genérico perfil, no qual todos os outros perfis são construídos, fornece uma maneira de estabelecer e manter links seguros (canais) entre o mestre e os escravos. O outro perfis genéricos definem os conceitos básicos de troca de objetos e transmissões de áudio e vídeo esporte. Perfis de utilitários são amplamente usados para funções como emular um serial linha, que é especialmente útil para muitos aplicativos legados.

Era realmente necessário explicar todos esses aplicativos em detalhes e fornecer pilhas de protocolo diferentes para cada um? Provavelmente não, mas havia uma série de diferentes grupos de trabalho que conceberam diferentes partes do padrão, e cada um apenas focou em seu problema específico e gerou seu próprio perfil. Pense nisso como a Lei de Conway em ação. (Na edição de abril de 1968 da revista *Datamation* , Melvin Conway observou que se você designar n pessoas para escrever um compilador, você obter um compilador n -pass, ou mais geralmente, a estrutura do software espelha a estrutura do grupo que o produziu.) Provavelmente, seria possível obter afastado com duas pilhas de protocolo em vez de 25, uma para transferência de arquivos e outra para streaming de comunicação em tempo real.

4.6.3 A pilha de protocolo Bluetooth

O padrão Bluetooth tem muitos protocolos agrupados vagamente em camadas mostrado na Fig. 4-35. A primeira observação a fazer é que a estrutura não siga o modelo OSI, o modelo TCP / IP, o modelo 802 ou qualquer outro modelo. A camada inferior é a camada de rádio física, que corresponde muito bem a a camada física nos modelos OSI e 802. Trata da transmissão de rádio e modulação. Muitas das preocupações aqui têm a ver com o objetivo de tornar o sistema barato para que possa se tornar um item de mercado de massa.

Página 347

SEC. 4,6
BLUETOOTH

323

Host-controlador
interface
Superior
camadas
Link de dados
camada
Física
camada
Rádio
Controle de link
(Banda base)
Gerenciador de links
L2CAP
Serviço
descoberta
RFCOMM
Formulários
...
Perfil
Perfil
Perfil

Figura 4-35. A arquitetura do protocolo Bluetooth.

A camada de controle de link (ou banda de base) é um pouco análoga ao sub-camada, mas também inclui elementos da camada física. Trata de como o master controla os intervalos de tempo e como esses intervalos são agrupados em quadros. Em seguida, vêm dois protocolos que usam o protocolo de controle de link. O gerenciador de links lida com o estabelecimento de canais lógicos entre dispositivos, incluindo energia gerenciamento, emparelhamento e criptografia e qualidade de serviço. Encontra-se abaixo do hospedeiro linha de interface do controlador. Esta interface é uma conveniência para implementação: tipicamente, os protocolos abaixo da linha serão implementados em um chip Bluetooth, e os protocolos acima da linha serão implementados no dispositivo Bluetooth que hospeda o chip.

O protocolo de link acima da linha é **L2CAP (Logical Link Control Adaptador de aplicação)**. Ele enquadraria mensagens de comprimento variável e fornece confiabilidade se necessário. Muitos protocolos usam L2CAP, como os dois protocolos utilitários que são mostrando. O protocolo de descoberta de serviço é usado para localizar serviços dentro da rede trabalhos. O protocolo RFCOMM (comunicação de radiofrequência) emula o porta serial padrão encontrada em PCs para conectar o teclado, mouse e modem, entre outros dispositivos.

A camada superior é onde os aplicativos estão localizados. Os perfis são representados inseridos por caixas verticais porque cada um define uma fatia da pilha de protocolo para um propósito particular. Perfis específicos, como o perfil do fone de ouvido, geralmente contêm apenas os protocolos necessários para esse aplicativo e nenhum outro. Por exemplo, os arquivos podem incluir L2CAP se eles tiverem pacotes para enviar, mas pular L2CAP se eles tem apenas um fluxo constante de amostras de áudio.

Nas seções a seguir, examinaremos a camada de rádio Bluetooth e vários nossos protocolos de link, uma vez que estes correspondem aproximadamente ao físico e ao MAC subcamadas nas outras pilhas de protocolo que estudamos.

324

O SUBCAMADA DE CONTROLE DE ACESSO MÉDIO INDIVÍDUO. 4

4.6.4 A Camada de Rádio Bluetooth

A camada de rádio move os bits do mestre para o escravo ou vice-versa. É um sistema de baixa potência com alcance de 10 metros operando no mesmo ISM de 2,4 GHz banda como 802.11. A banda é dividida em 79 canais de 1 MHz cada. Para coexistir com outras redes que usam a banda ISM, o espectro de propagação de salto de frequência é usava. Pode haver até 1600 saltos / s em slots com um tempo de permanência de 625 µs. Todos os nós em uma frequência de salto de piconet simultaneamente, seguindo o intervalo de tempo

sequência de saltos pseudo-aleatória e ditada pelo mestre.

Infelizmente, descobriu-se que as primeiras versões do Bluetooth e 802.11 interferidos o suficiente para arruinar as transmissões um do outro. Algumas empresas responderam por banindo completamente o Bluetooth, mas eventualmente uma solução técnica foi concebida.

A solução é o Bluetooth adaptar sua sequência de saltos para excluir canais em que existem outros sinais de RF. Este processo reduz a interferência prejudicial.

E chamado **de salto de frequência adaptativo**.

Três formas de modulação são usadas para enviar bits em um canal. O básico esquema é usar o chaveamento de mudança de frequência para enviar um símbolo de 1 bit a cada microsegundo,

dando uma taxa de dados bruta de 1 Mbps. Taxas aprimoradas foram introduzidas com o 2.0 versão do Bluetooth. Essas taxas usam chaveamento de mudança de fase para enviar 2 ou 3 bits por símbolo, para taxas de dados brutas de 2 ou 3 Mbps. As taxas aprimoradas são usadas apenas na parte de dados dos quadros.

4.6.5 As camadas de link Bluetooth

A camada de controle de link (ou banda base) é a coisa mais próxima que o Bluetooth tem de um Subcamada MAC. Ele transforma o fluxo de bits brutos em quadros e define algumas chaves para esteiras. Na forma mais simples, o mestre em cada piconet define uma série de 625-slots de tempo µsec, com as transmissões do mestre começando nos slots pares e no transmissões de escravos começando nos ímpares. Este esquema é o tempo tradicional di-multiplexação de visão, com o mestre obtendo metade dos slots e os escravos compartilhando a outra metade. Os quadros podem ter 1, 3 ou 5 slots de comprimento. Cada quadro tem uma sobrecarga de

126 bits para um código de acesso e cabeçalho, mais um tempo de acomodação de 250-260 µs por salto para permitir que os circuitos de rádio baratos se tornem estáveis. A carga útil do quadro pode ser criptografado para confidencialidade com uma chave que é escolhida quando o mestre e escravo se conectam. Os saltos acontecem apenas entre os quadros, não durante um quadro. O resultado é que um quadro de 5 slots é muito mais eficiente do que um quadro de 1 slot porque a sobrecarga é constante, mas mais dados são enviados.

O protocolo do gerenciador de links configura canais lógicos, chamados **links**, para transportar quadros entre o mestre e um dispositivo escravo que se descobriram. UMA o procedimento de emparelhamento é seguido para se certificar de que os dois dispositivos têm permissão para

comunicar antes de o link ser usado. O método de pareamento antigo é que ambos os dispositivos deve ser configurado com o mesmo PIN de quatro dígitos (Personal Identification Number). O PIN correspondente é como cada dispositivo saberia que estava se conectando

SEC. 4,6
BLUETOOTH

325

o dispositivo remoto certo. No entanto, usuários e dispositivos sem imaginação usam como padrão PINs como " 0000 " e " 1234 " significavam que este método forneceu muito pouco segurança na prática.

O novo método de **emparelhamento simples e seguro** permite que os usuários confirmem que ambos

vícios estão exibindo a mesma senha ou para observar a senha em um dispositivo e insira-o no segundo dispositivo. Este método é mais seguro porque os usuários não precisa escolher ou definir um PIN. Eles meramente confirmam um mais longo, gerado pelo dispositivo

chave de acesso. Claro, não pode ser usado em alguns dispositivos com entrada / saída limitada, como um fone de ouvido viva-voz.

Assim que o emparelhamento estiver concluído, o protocolo do gerenciador de link configura os links. Dois

os principais tipos de links existem para transportar dados do usuário. O primeiro é o **SCO (Síncrono)**

Link **orientado para conexão** . É usado para dados em tempo real, como telefone com conexões. Este tipo de link é alocado em um slot fixo em cada direção. Um escravo pode ter até três links SCO com seu mestre. Cada link SCO pode transmitir um Canal de áudio PCM de 64.000 bps. Devido à natureza crítica dos links SCO, os quadros enviados por eles nunca são retransmitidos. Em vez disso, encaminhe a correção de erros pode ser usado para aumentar a confiabilidade.

O outro tipo é o link **ACL (Asynchronous ConnectionLess)**. Esse tipo de link é usado para dados comutados por pacote que estão disponíveis em intervalos irregulares. O tráfego ACL é entregue com base no melhor esforço. Nenhuma garantia é dada. Molduras pode ser perdido e pode ter que ser retransmitido. Um escravo pode ter apenas uma ACL link para seu mestre.

Os dados enviados por links ACL vêm da camada L2CAP. Esta camada tem quatro funções principais. Primeiro, ele aceita pacotes de até 64 KB da camada superior e os divide em quadros para transmissão. No final, os frames são remontados em pacotes. Em segundo lugar, ele lida com a multiplexação e demultiplexação de múltiplas fontes de pacotes. Quando um pacote foi remontado, o L2CAP camada determina qual protocolo de camada superior deve ser entregue, por exemplo, RFcomm ou descoberta de serviço. Terceiro, o L2CAP trata do controle de erros e da retransmissão. Isto detecta erros e reenvia pacotes que não foram reconhecidos. Finalmente, L2CAP impõe requisitos de qualidade de serviço entre vários links.

4.6.6 A Estrutura do Frame Bluetooth

Bluetooth define vários formatos de quadro, o mais importante deles é mostrado em duas formas na Fig. 4-36. Ele começa com um código de acesso que geralmente identifica localiza o mestre para que os escravos dentro do alcance de rádio de dois mestres possam dizer qual tráfego fic é para eles. Em seguida, vem um cabeçalho de 54 bits contendo uma subcamada MAC típica Campos. Se o quadro for enviado na taxa básica, o campo de dados vem a seguir. Tem até 2744 bits para uma transmissão de cinco slots. Para um único intervalo de tempo, o formato é o mesmo, exceto que o campo de dados é de 240 bits. Se o quadro for enviado na taxa aprimorada, a porção de dados pode ter até dois ou três vezes mais bits porque cada símbolo carrega 2 ou 3 bits em vez de 1

```

Guarda / Sincronização
Reboque
Bits
72
54
16
0-8184
2
(b) Quadro de dados de taxa aprimorada, inferior
5 x 675 slots de microseg
Endereço Tipo FAS CRC
3
4
1 1 1
8

```

Figura 4-36. Frame de dados Bluetooth típico em taxas de dados (a) básicas e (b) aprimoradas.
 morreu. Esses dados são precedidos por um campo de guarda e um padrão de sincronização que é usado para mudar para a taxa de dados mais rápida. Ou seja, o código de acesso e o cabeçalho são car-

ried na taxa básica e apenas a parte dos dados é transportada na taxa mais rápida.

Os quadros de taxa aprimorada terminam com um pequeno trailer.

Vamos dar uma olhada rápida no cabeçalho comum. O campo *Endereço* identifica para qual dos oito dispositivos ativos o quadro se destina. O campo *Tipo* identifica localiza o tipo de frame (ACL, SCO, poll ou nulo), o tipo de correção de erro usado em o campo de dados e quantos slots o quadro tem. O bit de *fluxo* é afirmado por um escravo quando seu buffer está cheio e não pode receber mais dados. Este bit permite um forma primitiva de controle de fluxo. O bit de *reconhecimento* é usado para carregar uma ACK em um quadro. O bit de *sequência* é usado para numerar os quadros para detectar retransmissões. O protocolo é parar e esperar, então 1 bit é suficiente. Então vem o Cabeçalho de 8 bits *Checksum*. Todo o cabeçalho de 18 bits é repetido três vezes para formar o cabeçalho de 54 bits mostrado na Figura 4-36. Do lado receptor, um circuito simples examina todas as três cópias de cada bit. Se todos os três forem iguais, o bit é aceito. E se não, a opinião da maioria vence. Assim, 54 bits de capacidade de transmissão são usados para enviar 10 bits de cabeçalho. A razão é que, para enviar dados de forma confiável em um ambiente barulhento,

uso de dispositivos baratos e de baixa potência (2,5 mW) com pouca capacidade de computação, uma grande quantidade de redundância é necessária.

Vários formatos são usados para o campo de dados para quadros ACL e SCO. Os quadros SCO de taxa básica são um exemplo simples de estudar: o campo de dados é sempre 240 bits. Três variantes são definidas, permitindo 80, 160 ou 240 bits de carga útil real, com o resto sendo usado para correção de erros. Na versão mais confiável (80 bits carga útil), o conteúdo é repetido apenas três vezes, o mesmo que o cabeçalho. Podemos calcular a capacidade com este quadro da seguinte maneira. Já que o escravo pode usar apenas os slots ímpares, ele obtém 800 slots / s, assim como o mestre. Com um 80 bits

carga útil, a capacidade do canal do escravo é 64.000 bps, assim como o canal capacidade do mestre. Esta capacidade é exatamente o suficiente para um único full-duplex Canal de voz PCM (razão pela qual foi escolhida uma taxa de salto de 1600 saltos / s). que é, apesar de uma largura de banda bruta de 1 Mbps, uma única voz full-duplex não compactada canal pode saturar completamente a piconet. A eficiência de 13% é o resultado de gastar 41% da capacidade em tempo de acomodação, 20% em cabeçalhos e 26% em codificação de repetição. Esta deficiência destaca o valor das taxas aprimoradas e quadros de mais de um único slot.

Há muito mais a ser dito sobre o Bluetooth, mas não há mais espaço para dizê-lo aqui. Para os curiosos, a especificação Bluetooth 4.0 contém todos os detalhes.

4.7 RFID

Vimos designs de MAC de LANs a MANs e até PANs.

Como último exemplo, estudaremos uma categoria de dispositivos sem fio de baixo custo que as pessoas

pode não reconhecer como formando uma rede de computadores: o **RFID (Radio Frequency Identification de frequência)** e leitores que descrevemos na Seç. 1.5.4.

A tecnologia RFID assume muitas formas, usadas em cartões inteligentes, implantes para animais de estimação,

passaportes, livros da biblioteca e muito mais. O formulário que veremos foi desenvolvido na busca por um **EPC (Código Eletrônico de Produto)** que começou com o Auto-ID Center no Massachusetts Institute of Technology em 1999. Um EPC é um re-colocação de um código de barras que pode transportar uma grande quantidade de informações e é elec-

legível tronicamente em distâncias de até 10 m, mesmo quando não é visível. É dif-tecnologia diferente do que, por exemplo, o RFID usado em passaportes, que devem ser colocado bem perto de um leitor para realizar uma transação. A capacidade de comunicar

cate à distância torna as CEPs mais relevantes para nossos estudos.

EPCglobal foi formada em 2003 para comercializar o desenvolvimento de tecnologia RFID operado pelo Auto-ID Center. O esforço ganhou um impulso em 2005, quando o Walmart voltou a exigiu que seus 100 principais fornecedores etiquetassem todas as remessas com etiquetas

RFID. Difundido

implantação tem sido dificultada pela dificuldade de competir com impressos baratos códigos de barras, mas novos usos, como em carteiras de motorista, agora estão crescendo. Nós iremos de-

escriba a segunda geração desta tecnologia, que é informalmente chamada de **EPC Gen 2** (EPCglobal, 2008).

4.7.1 Arquitetura EPC Gen 2

A arquitetura de uma rede RFID EPC Gen 2 é mostrada na Figura 4-37. Tem dois componentes principais: tags e leitores. As etiquetas RFID são dispositivos pequenos e baratos que têm um identificador EPC exclusivo de 96 bits e uma pequena quantidade de memória que pode ser lida e escrita pelo leitor RFID. A memória pode ser usada para registrar o histórico de localização de um item, por exemplo, conforme ele se move pela cadeia de suprimentos.

Página 352

328

O SUBCAMADA DE CONTROLE DE ACESSO MÉDIO
INDIVÍDUO. 4

Muitas vezes, as tags parecem adesivos que podem ser colocados, por exemplo, em pares de jeans nas prateleiras de uma loja. A maior parte do adesivo é captado por uma antena que é impresso nele. Um minúsculo ponto no meio é o circuito integrado RFID. Alternativamente, as etiquetas RFID podem ser integradas a um objeto, como uma carteira de motorista. No ambos os casos, as etiquetas não têm bateria e devem obter energia do rádio transmissões de um leitor RFID próximo para funcionar. Este tipo de tag é chamado de " Classe 1 " tag para distingui-lo de tags mais capazes que têm baterias.

RFID
leitor
RFID
etiqueta
Backscatter
sinal
Leitor
sinal

Figura 4-37. Arquitetura RFID.

Os leitores são a inteligência do sistema, análoga às estações base e pontos de acesso em redes celulares e sem fio. Os leitores são muito mais poderosos do que tags. Eles têm suas próprias fontes de energia, muitas vezes têm várias antenas e são responsáveis por quando as tags enviam e recebem mensagens. Como normalmente ser várias tags dentro da faixa de leitura, os leitores devem resolver os vários ac-problema cesso. Pode haver vários leitores que podem competir uns com os outros em a mesma área também.

A principal tarefa do leitor é inventariar as tags na vizinhança, que é, para descobrir os identificadores das tags próximas. O inventário é realizado

com o protocolo da camada física e o protocolo de identificação de tag que estão fora alinhado nas seguintes seções.

4.7.2 Camada Física EPC Gen 2

A camada física define como os bits são enviados entre o leitor RFID e Tag. Muito dele usa métodos para enviar sinais sem fio que vimos antes vily. Nos EUA, as transmissões são enviadas no ISM 902-928 MHz não licenciado banda. Esta banda está na faixa de UHF (Ultra High Frequency), então as tags são referidas como etiquetas RFID UHF. O leitor executa salto de frequência pelo menos a cada 400 ms para espalhar seu sinal pelo canal, para limitar a interferência e satisfazer os requisitos regulamentares. O leitor e as tags usam formas de ASK (Amplitude Shift Keying) modulação que descrevemos na Seç. 2.5.2 para codificar bits. Eles se revezam para enviar bits, de modo que o link é half duplex.

Página 353

SEC. 4,7

RFID

329

Existem duas diferenças principais de outras camadas físicas que temos Ied. A primeira é que o leitor está sempre transmitindo um sinal, independentemente de se é o leitor ou tag que está se comunicando. Naturalmente, o leitor transmite um sinal para enviar bits para tags. Para que as tags enviem bits ao leitor, o leitor transmite um sinal de portadora fixa sem bits. As tags coletam este sinal para obter a energia de que precisam para funcionar; caso contrário, uma tag não seria capaz de transmitir em

o primeiro lugar. Para enviar dados, uma tag muda se está refletindo o sinal de o leitor, como um sinal de radar ricocheteando em um alvo ou absorvendo-o.

Este método é denominado **retroespalhamento**. É diferente de todas as outras situações sem fio que vimos até agora, em que o remetente e o receptor nunca transmitem em o mesmo tempo. O retroespalhamento é uma forma de baixa energia para a tag criar um sinal fraco próprio que aparece no leitor. Para o leitor decodificar o sinal, ele deve filtrar o sinal de saída que está transmitindo. Porque a tag o sinal é fraco, as tags só podem enviar bits para o leitor a uma taxa baixa e as tags não podem receber ou mesmo detectar transmissões de outras tags.

A segunda diferença é que formas muito simples de modulação são usadas para que eles podem ser implementados em uma tag que funciona com muito pouca energia e custa apenas um

poucos centavos para fazer. Para enviar dados aos tags, o leitor usa dois níveis de amplitude. Os bits são determinados como 0 ou 1, dependendo de quanto tempo o leitor espera antes de um período de baixa potência. A tag mede o tempo entre baixa potência períodos e compara este tempo com uma referência medida durante um preâmbulo. Como mostrado na Figura 4-38, 1s são mais longos que 0s.

As respostas da tag consistem na tag alternando seu estado de retroespalhamento em intervalos para criar uma série de pulsos no sinal. Em qualquer lugar de um a oito pulsos períodos podem ser usados para codificar cada 0 ou 1, dependendo da necessidade de confiabilidade. 1s têm menos transições do que 0s, como é mostrado com um exemplo de dois pulsos codificação de período na Fig. 4-38.

Tempo
Poder
Leitor
“0”
Leitor
“1”
Tag
“0”
Tag
“1”
Backscatter

Figura 4-38. Leitor e sinais de retroespalhamento de tag.

4.7.3 Camada de Identificação de Tag EPC Gen 2

Para inventariar as tags próximas, o leitor precisa receber uma mensagem de cada tag que fornece o identificador da tag. Esta situação é um problema de acesso múltiplo

para o qual o número de tags é desconhecido no caso geral. O leitor pode

330

O SUBCAMADA DE CONTROLE DE ACESSO MÉDIO INDIVÍDUO. 4

transmita uma consulta para solicitar que todas as tags enviem seus identificadores. No entanto, as tags que re-dobrado imediatamente, então colidiria da mesma maneira que as estações em um clássico Ethernet.

Vimos muitas maneiras de lidar com o problema de acesso múltiplo neste capítulo. O protocolo mais próximo para a situação atual, em que as tags não podem ouvir as transmissões uns dos outros, é ALOHA entalhado, um dos primeiros protocolos. Nós estudamos. Este protocolo é adaptado para uso em Gen 2 RFID.

A seqüência de mensagens usadas para identificar uma tag é mostrada na Figura 4-39. No primeiro slot (slot 0), o leitor envia uma mensagem de *consulta* para iniciar o processo. Cada *A* mensagem *QRepeat* avança para o próximo slot. O leitor também diz às tags que faixa de slots sobre a qual randomizar as transmissões. Usar um intervalo é necessário porque o leitor sincroniza as tags ao iniciar o processo; ao contrário das estações em uma Ethernet, os tags não despertam com uma mensagem no momento de sua escolha.

Tempo
Etiqueta RFID
Consulta (slot 0)
RN16 (slot 2)
Identificador EPC
.

.

QRepeat (slot1)
Ack
QRepeat (slot 2)
QRepeat (slot N)
QRepeat (slot 3)
Leitor RFID

Figura 4-39. Exemplo de troca de mensagens para identificar uma tag.

As tags escolhem um espaço aleatório para responder. Na Fig. 4-39, a tag responde no slot 2. No entanto, as tags não enviam seus identificadores quando respondem pela primeira vez. Em vez disso, uma tag envia um número aleatório curto de 16 bits em uma mensagem *RN16*. Se não houver colisão, o leitor recebe esta mensagem e envia uma mensagem *ACK* própria. Neste estágio, a tag adquiriu o slot e envia seu identificador *EPC*. A razão para essa troca é que os identificadores *EPC* são longos, então as colisões dessas mensagens seriam caras. Em vez disso, uma pequena troca é usada para testar se a tag pode usar com segurança o slot para enviar seu identificador. Assim que seu identificador tiver transmitido com sucesso, a tag para temporariamente de responder à nova *consulta* mensagens para que todas as tags restantes possam ser identificadas.

SEC. 4,7

RFID

331

Um problema chave é o leitor ajustar o número de slots para evitar colisões ions, mas sem usar tantos slots que o desempenho é prejudicado. Este ajuste é análogo ao backoff exponencial binário na Ethernet. Se o leitor vir muitos slots sem respostas ou muitos slots com colisões, ele pode enviar um *QAdjust* mensagem para diminuir ou aumentar o intervalo de slots sobre os quais as tags são responja.

O leitor RFID pode realizar outras operações nas etiquetas. Por exemplo, pode selecionar um subconjunto de tags antes de executar um inventário, permitindo que ele colete

patrocínios de, digamos, jeans etiquetados, mas não camisas etiquetadas. O leitor também pode escrever dados para tags à medida que são identificados. Este recurso pode ser usado para registrar o ponto de venda ou outras informações relevantes.

4.7.4 Formatos de mensagem de identificação de tag

O formato da mensagem de *consulta* é mostrado na Figura 4-40 como um exemplo de um mensagem do leitor para tag. A mensagem é compacta porque as taxas de downlink são limitado, de 27 kbps a 128 kbps. O campo *Command* carrega o código 1000 para identificar a mensagem como uma *consulta*.

Parâmetros físicos

2
4
5
2
4
1
1

Bits

Comando

1000

DR

M

TR

2

1

Sessão

Alvo

Sel

Q

CRC

Seleção de tag

Figura 4-40. Formato da mensagem de consulta.

Os próximos sinalizadores, *DR*, *M* e *TR*, determinam os parâmetros da camada física para transmissões de leitores e respostas de tags. Por exemplo, a taxa de resposta pode ser definida entre 5 kbps e 640 kbps. Iremos pular os detalhes dessas bandeiras.

Em seguida, vêm três campos, *Sel*, *Sessão* e *Destino*, que selecionam as tags a serem respondidas. Assim como os leitores podem selecionar um subconjunto de identificadores, as tags acompanham até quatro sessões simultâneas e se elas foram identificadas nessas sessões. Desta forma, vários leitores podem operar em sobreposição de capa áreas de idade usando sessões diferentes.

Em seguida é o parâmetro mais importante para este comando, *Q*. Este campo define a faixa de slots sobre os quais as tags responderão, de 0 a $2^Q - 1$. Finalmente, existe um CRC para proteger os campos da mensagem. Com 5 bits, é mais curto do que a maioria dos CRCs que temos

visto, mas a mensagem de *consulta* é muito mais curta do que a maioria dos pacotes também. As mensagens tag-to-reader são mais simples. Uma vez que o leitor está no controle, ele sabe que mensagem esperar em resposta a cada uma de suas transmissões. A marca de patrocinadores simplesmente carregam dados, como o identificador EPC.

Página 356

332

O SUBCAMADA DE CONTROLE DE ACESSO MÉDIO INDIVÍDUO. 4

Originalmente, as etiquetas eram apenas para fins de identificação. No entanto, eles têm crescido com o tempo para se parecer com computadores muito pequenos. Algumas tags de pesquisa têm sensores e são capazes de executar pequenos programas para coletar e processar dados (Sample et al., 2008). Uma visão para esta tecnologia é a " Internet das coisas " que conecta objetos no mundo físico para a Internet (Welbourne et al., 2009; e Gershenson et al., 2004).

4.8 TROCA DA CAMADA DO LINK DE DADOS

Muitas organizações têm várias LANs e desejam conectá-las. Seria conveniente se pudéssemos apenas juntar as LANs para fazer uma LAN maior? Na verdade, podemos fazer isso quando as conexões são feitas com dispositivos chamados

pontes. Os switches Ethernet que descrevemos na Seç. 4.3.4 são um nome moderno para pontes; eles fornecem funcionalidade que vai além da clássica Ethernet e Ethernet hubs para facilitar a união de várias LANs em uma rede maior e mais rápida. Nós deve usar os termos "ponte" e "switch" de forma intercambiável.

As bridges operam na camada de enlace de dados, então examinam a camada de enlace de dados e vestidos para encaminhar armações. Uma vez que não devem examinar a carga útil campo dos frames que encaminham, eles podem lidar com pacotes IP, bem como outros tipos de pacotes, como pacotes AppleTalk. Em contraste, os *roteadores* examinam os endereços em pacotes e rotear com base neles, então eles só funcionam com os protocolos que eles foram projetados para lidar com.

Nesta seção, veremos como as pontes funcionam e são usadas para unir vários LANs físicas em uma única LAN lógica. Também veremos como fazer o rever e tratar uma LAN física como várias LANs lógicas, chamadas **VLANs (Virtual LANs reais)**. Ambas as tecnologias fornecem flexibilidade útil para o gerenciamento de redes. Para um tratamento abrangente de pontes, switches e tópicos relacionados, consulte Seifert e Edwards (2008) e Perlman (2000).

4.8.1 Usos de pontes

Antes de entrar na tecnologia de pontes, vamos dar uma olhada em algumas várias situações em que são utilizadas pontes. Mencionaremos três razões pelas quais um uma única organização pode acabar com várias LANs.

Primeiro, muitos departamentos universitários e corporativos têm suas próprias LANs para conectar seus próprios computadores pessoais, servidores e dispositivos, como impressoras. Uma vez que os objetivos dos vários departamentos são diferentes, diferentes departamentos podem definir diferentes LANs, sem levar em conta o que os outros departamentos estão fazendo. Mais cedo ou mais tarde, porém, há uma necessidade de interação, portanto, são necessárias pontes. Neste exemplo, várias LANs passam a existir devido à autonomia de seus proprietários.

Página 357

SEC. 4,8

COMUTAÇÃO DA CAMADA DE LINK DE DADOS

333

Em segundo lugar, a organização pode estar geograficamente distribuída por vários edifícios separados por distâncias consideráveis. Pode ser mais barato ter LANs separadas em cada edifício e conectá-los com pontes e algumas fibras ópticas de longa distância links do que passar todos os cabos para um único switch central. Mesmo se colocando os cabos é fácil de fazer, há limites em seus comprimentos (por exemplo, 200 m para gigabit de par trançado Ethernet). A rede não funcionaria para cabos mais longos devido ao excesso atenuação do sinal ou atraso de ida e volta. A única solução é partitionar a LAN e instalar pontes para juntar as peças para aumentar a distância física total que pode ser coberto.

Terceiro, pode ser necessário dividir o que é logicamente uma única LAN em separado LANs de taxa (conectadas por pontes) para acomodar a carga. Em muitas grandes unidades versidades, por exemplo, milhares de estações de trabalho estão disponíveis para alunos e computação docente. As empresas também podem ter milhares de funcionários. A escala deste sistema impede colocar todas as estações de trabalho em uma única LAN - há mais computadores do que portas em qualquer hub Ethernet e mais estações do que o permitido em uma única Ethernet clássica.

Mesmo se fosse possível conectar todas as estações de trabalho juntas, colocando mais estações em um hub Ethernet ou Ethernet clássica não aumentariam a capacidade. Todos os as estações compartilham a mesma quantidade fixa de largura de banda. Quanto mais estações houver,

a menor largura de banda média por estação.

No entanto, duas LANs separadas têm o dobro da capacidade de uma única LAN.

As pontes permitem que as LANs sejam unidas enquanto mantém essa capacidade. A chave é não enviar tráfego para portas onde não é necessário, de modo que cada LAN possa funcionar em

velocidade máxima. Este comportamento também aumenta a confiabilidade, uma vez que em uma única LAN um defeito

nó ativo que mantém a saída de um fluxo contínuo de lixo pode obstruir o pneu LAN. Ao decidir o que encaminhar e o que não encaminhar, as pontes agem como portas corta-fogo em um edifício, evitando que um único nó que enlouqueceu de trazer desativando todo o sistema.

Para tornar esses benefícios facilmente disponíveis, o ideal é que as pontes sejam completamente transparente. Deve ser possível sair e comprar pontes, conectar os cabos LAN nas pontes e tudo funcionando perfeitamente, instantaneamente. Deveria haver nenhuma mudança de hardware necessária, nenhuma mudança de software necessária, nenhuma configuração de endereço interruptores, nenhum download de tabelas ou parâmetros de roteamento, nada. Basta ligar nos cabos e vá embora. Além disso, a operação das LANs existentes não deve ser afetado pelas pontes. No que diz respeito às estações, não deve haver nenhuma diferença observável se eles são ou não parte de uma ponte LAN. Deve ser tão fácil mover estações ao redor da LAN com ponte quanto é movê-los em uma única LAN.

Surpreendentemente, é realmente possível criar pontes transparentes ent. Dois algoritmos são usados: um algoritmo de aprendizagem reversa para impedir que o tráfego seja enviado onde não é necessário; e um algoritmo de spanning tree para quebrar loops que podem ser formada quando os interruptores são cabeados à toa. Vamos agora dar uma olhada a esses algoritmos, por sua vez, para aprender como essa mágica é realizada.

Página 358

334

O SUBCAMADA DE CONTROLE DE ACESSO MÉDIO INDIVÍDUO. 4

4.8.2 Pontes de Aprendizagem

A topologia de duas LANs interligadas é mostrada na Figura 4-41 para duas casos. No lado esquerdo, duas LANs multidrop, como Ethernets clássicas, são unido por uma estação especial - a ponte - que fica em ambas as LANs. À direita lado, LANs com cabos ponto a ponto, incluindo um hub, são unidos. o pontes são os dispositivos aos quais as estações e o hub estão conectados. Se a LAN tecnologia é Ethernet, as pontes são mais conhecidas como switches Ethernet.

(uma)

(b)

UMA

D

Ponte

B1

1

2

Porta

B

C

E

G

F

C

Ponte

B1

B2

UMA

B

G

D

H1

Porta

1

2

1

3

4

2

3

4

F

Figura 4-41. (a) Faça uma ponte conectando duas LANs multidrop. (b) Pontes (e um hub) conectando sete estações ponto a ponto.

As pontes foram desenvolvidas quando as Ethernets clássicas estavam em uso, por isso são frequentemente

mostrado em topologias com cabos multidrop, como na Fig. 4-41 (a). No entanto, todos os topologias que são encontradas hoje são compostas de cabos ponto a ponto e comuta. As pontes funcionam da mesma maneira em ambos os ambientes. Todas as estações em anexados à mesma porta em uma ponte pertencem ao mesmo domínio de colisão, e este é diferente do domínio de colisão para outras portas. Se houver mais de uma estação como em uma Ethernet clássica, um hub ou um link half-duplex, o protocolo CSMA / CD é usado para enviar frames.

Há uma diferença, entretanto, em como as LANs com ponte são construídas. Para fazer a ponte LANs multidrop, uma ponte é adicionada como uma nova estação em cada uma das LANs, como na Figura 4-41 (a). Para conectar LANs ponto a ponto, os hubs são ou conectado a uma ponte ou, de preferência, substituído por uma ponte para aumentar o desempenho. Na Figura 4-41 (b), as pontes substituíram todos, exceto um hub.

Diferentes tipos de cabos também podem ser conectados a uma ponte. Por exemplo, o cabo que conecta a ponte $B1$ à ponte $B2$ na Fig. 4-41 (b) pode ser de longa distância link de fibra óptica, enquanto o cabo que conecta as pontes às estações pode ser um linha de par trançado de curta distância. Este arranjo é útil para conectar LANs em diferentes edifícios diferentes.

Agora vamos considerar o que acontece dentro das pontes. Cada ponte opera em modo promíscuo, ou seja, aceita todos os frames transmitidos pelas estações

Página 359

SEC. 4,8

COMUTAÇÃO DA CAMADA DE LINK DE DADOS

335

anexado a cada uma de suas portas. A ponte deve decidir se deve avançar ou descartar cardar cada quadro e, se for o primeiro, em qual porta enviar o quadro. Esta decisão é feito usando o endereço de destino. Como exemplo, considere o topo- logia da Figura 4-41 (a). Se a estação A enviar um quadro para a estação B , a ponte $B1$ receberá o quadro na porta 1. Este quadro pode ser descartado imediatamente sem mais delongas porque já está na porta correta. No entanto, na topologia da Fig. 4-41 (b) suponha que A envia um quadro para D . A ponte $B1$ receberá o quadro na porta 1 e a saída na porta 4. A ponte $B2$ então receberá o quadro em sua porta 4 e fará a saída em sua porta 1.

Uma maneira simples de implementar este esquema é ter uma grande tabela (hash) dentro do ponte. A tabela pode listar cada destino possível e qual porta de saída deve ser anseia. Por exemplo, na Figura 4-41 (b), a tabela em $B1$ listaria D como pertencente à porta 4, uma vez que todos $B1$ tem que saber é qual a porta para colocar quadros em chegar a D . Que, na verdade, mais encaminhamento acontecerá mais tarde, quando o frame atingir $B2$ não é de interesse para $B1$.

Quando as pontes são conectadas pela primeira vez, todas as tabelas de hash estão vazias. Nenhum as pontes sabem onde qualquer um dos destinos estão, então eles usam um algoritmo de inundação ritmo: cada quadro de entrada para um destino desconhecido é enviado para todas as portas ao qual a ponte está conectada, exceto aquela em que ela chegou. Conforme o tempo passa, as pontes aprendem onde estão os destinos. Uma vez que um destino é conhecido, frames destinados a isso são colocados apenas na porta adequada; eles não são inundados.

O algoritmo usado pelas pontes é o **aprendizado retroativo**. Como mencionado acima, as pontes operam em modo promíscuo, para que eles vejam todos os quadros enviados qualquer uma de suas portas. Olhando para os endereços de origem, eles podem dizer quais má- chines são acessíveis em quais portas. Por exemplo, se a ponte $B1$ na Fig. 4-41 (b) vê um quadro na porta 3 vindo de C , ele sabe que C deve ser acessível através da porta 3, portanto, ele cria uma entrada em sua tabela de hash. Qualquer quadro subsequente endereçado a C

entrando em $B1$ em qualquer outra porta será encaminhado para a porta 3. A topologia pode mudar conforme as máquinas e pontes são ligadas e desligadas e mudou-se. Para lidar com topologias dinâmicas, sempre que uma entrada da tabela hash é feito, o tempo de chegada do quadro é anotado na entrada. Sempre que uma moldura cuja fonte já está na tabela chega, sua entrada é atualizada com o atual Tempo. Assim, o tempo associado a cada entrada diz a última vez que um quadro de aquela máquina foi vista.

Periodicamente, um processo na ponte verifica a tabela de hash e limpa todas as entradas com mais de alguns minutos. Desta forma, se um computador for desconectado de seu LAN, movido ao redor do edifício, e conectado novamente em outro lugar, dentro de um alguns minutos estará de volta ao funcionamento normal, sem qualquer intervenção manual. Este algoritmo também significa que se uma máquina ficar silenciosa por alguns minutos, qualquer tráfego enviado para ele terá de ser inundado até que ele envie um quadro em seguida. O procedimento de roteamento para um quadro de entrada depende da porta em que ele chega (a porta de origem) e o endereço ao qual se destina (o endereço de destino). O procedimento é o seguinte.

Página 360

336

O SUBCAMADA DE CONTROLE DE ACESSO MÉDIO INDIVÍDUO. 4

1. Se a porta do endereço de destino for igual à porta de origem, descarte a moldura.
2. Se a porta para o endereço de destino e a porta de origem forem diferentes, encaminhe o quadro para a porta de destino.
3. Se a porta de destino for desconhecida, use flooding e envie o quadro em todas as portas, exceto a porta de origem.

Você pode se perguntar se o primeiro caso pode ocorrer com links ponto a ponto. A resposta é que pode ocorrer se os hubs forem usados para conectar um grupo de computadores a um ponte. Um exemplo é mostrado na Figura 4-41 (b), onde as estações E e F estão conectadas ao hub $H1$, que por sua vez está conectado à ponte $B2$. Se E envia um quadro para F , o hub retransmiti-lo para $B2$, bem como a F . É isso que os hubs fazem - conectam todas as portas juntas para que uma entrada de quadro em uma porta seja simplesmente enviada para todas as outras portas. o

quadro chegará em $B2$ na porta 4, que já é a porta de saída certa para alcançar o destino. A ponte $B2$ precisa apenas descartar o quadro.

Conforme cada quadro chega, este algoritmo deve ser aplicado, por isso é geralmente implementado com chips VLSI de uso especial. Os chips fazem a pesquisa e atualizam o entrada na tabela, tudo em alguns microssegundos. Porque as pontes olham apenas para o anúncio MAC

vestidos para decidir como encaminhar armadilhas, é possível começar a encaminhar assim que conforme o campo de cabeçalho de destino chega, antes que o resto do quadro chegue (desde que a linha de saída esteja disponível, é claro). Este design reduz a latência de passagem pela ponte, bem como o número de frames que a ponte deve ser capaz de armazenar em buffer. É referido como **comutação cut-through** ou **buraco de minhoca**

roteamento e geralmente é tratado em hardware.

Podemos olhar para a operação de uma ponte em termos de pilhas de protocolo para entender o que significa ser um dispositivo de camada de link. Considere um quadro enviado da estação A

para a estação D na configuração da Figura 4-41 (a), na qual as LANs são Ethernet. O quadro passará por uma ponte. A visão da pilha de protocolo de processamento é mostrado na Fig. 4-42.

O pacote vem de uma camada superior e desce para o MAC Ethernet camada. Ele adquire um cabeçalho Ethernet (e também um trailer, não mostrado na figura). Esta unidade é passada para a camada física, passa pelo cabo e é recolhida

pela ponte.

Na ponte, o quadro é passado da camada física para a Ethernet

Camada MAC. Esta camada estendeu o processamento em comparação com o MAC Ethernet camada em uma estação. Ele passa o quadro para um relé, ainda dentro da camada MAC. A função de relé de ponte usa apenas o cabeçalho Ethernet MAC para determinar como lidar com o quadro. Neste caso, ele passa o quadro para a camada Ethernet MAC de uma porta costumava chegar à estação D e o quadro continua seu caminho.

No caso geral, relés em uma determinada camada podem reescrever os cabeçalhos para aquele camada. As VLANs fornecerão um exemplo em breve. Em nenhum caso a ponte deve parecer dentro do quadro e aprenda que ele está carregando um pacote IP; isso é irrelevante para o

Página 361

SEC. 4,8

COMUTAÇÃO DA CAMADA DE LINK DE DADOS

337

Eth
Eth
Pacote
Pacote
Pacote
Pacote
Retransmissão
Rede
Ethernet
MAC
Física
Ponte
Estação D
Estação A
Fio
Fio
Eth
Eth
Pacote
Pacote
Pacote
Eth Packet
Eth Packet
Eth Packet
Eth Packet

Figura 4-42. Processamento de protocolo em uma ponte.

o processamento da ponte e violaria as camadas de protocolo. Observe também que uma ponte com k portas terá k instâncias de MAC e camadas físicas. O valor de k é 2 para nosso exemplo simples.

4.8.3 Spanning Tree Bridges

Para aumentar a confiabilidade, links redundantes podem ser usados entre pontes. No

No exemplo da Figura 4-43, existem dois links em paralelo entre um par de pontes.

Este design garante que se um link for cortado, a rede não será particionada em dois conjuntos de computadores que não podem se comunicar.

Quadro F₀
Ponte
B1
UMA
B2
Links redundantes
F₁
F₂
F₃
F₄

Figura 4-43. Pontes com dois links paralelos.

No entanto, essa redundância apresenta alguns problemas adicionais porque cria loops na topologia. Um exemplo desses problemas pode ser visto por em como um quadro enviado por A para um destino não observado anteriormente é tratado em Fig. 4-43. Cada ponte segue a regra normal para lidar com destinos desconhecidos ções, que deve inundar o quadro. Chame o quadro de A que atinge a ponte B1 quadro F₀. A ponte envia cópias deste quadro por todas as suas outras portas. Nós

Página 362

338

O SUBCAMADA DE CONTROLE DE ACESSO MÉDIO INDIVÍDUO. 4

irá considerar apenas as portas de ponte que conectam B_1 a B_2 (embora o quadro seja enviado para as outras portas também). Uma vez que existem dois links de B_1 a B_2 , duas cópias do quadro alcançará B_2 . Eles são mostrados na Fig. 4-43 como F_1 e F_2 .

Pouco depois, a ponte B_2 recebe esses quadros. No entanto, não (e não pode) saber que são cópias do mesmo quadro, ao invés de dois diferentes quadros enviados um após o outro. Então, a ponte B_2 pega F_1 e envia cópias dela todas as outras portas, e também pega F_2 e envia cópias para todas as outras portas. Isso produz os quadros F_3 e F_4 que são enviados ao longo dos dois links de volta para B_1 . A ponte B_1 então vê dois novos quadros com destinos e cópias desconhecidos eis de novo. Este ciclo continua para sempre.

A solução para esta dificuldade é que as pontes se comuniquem com cada outro e sobrepor a topologia real com uma árvore de abrangência que atinge todos ponte. Com efeito, algumas conexões potenciais entre as pontes são ignoradas no interesse de construir uma topologia fictícia sem loops que seja um subconjunto do real topologia.

Por exemplo, na Figura 4-44, vemos cinco pontes que estão interconectadas e também têm estações conectadas a eles. Cada estação se conecta a apenas uma ponte. Lá são algumas conexões redundantes entre as pontes para que os quadros sejam para-guardados em loops se todos os links forem usados. Esta topologia pode ser considerada como um gráfico em que as pontes são os nós e os links ponto a ponto são os arestas. O gráfico pode ser reduzido a uma árvore geradora, que não tem ciclos por definição, eliminando os links mostrados como linhas tracejadas na Figura 4.44. Usando esta extensão-Nesta árvore, há exatamente um caminho de cada estação para cada outra estação. Uma vez as pontes concordaram com a árvore de abrangência, todos encaminhando entre as estações seguindo baixa a árvore geradora. Uma vez que existe um caminho único de cada fonte para cada destino, os loops são impossíveis.

Ponte
Estação
 B_1
 B_2
 B_3
 B_4
 B_5
Link que não faz parte
da árvore geradora
Raiz
ponte

Figura 4-44. Uma árvore que conecta cinco pontes. As linhas tracejadas são links que não fazem parte da árvore de abrangência.

Para construir a árvore estendida, as pontes executam um algoritmo distribuído. Cada ponte periodicamente transmite uma mensagem de configuração de todas as suas portas para o seu

SEC. 4,8

COMUTAÇÃO DA CAMADA DE LINK DE DADOS

339

vizinhos e processa as mensagens que recebe de outras pontes, conforme descrito Próximo. Essas mensagens não são encaminhadas, pois têm como objetivo construir a árvore, que pode então ser usado para encaminhamento.

As pontes devem primeiro escolher uma ponte para ser a raiz da árvore estendida. Para fazer essa escolha, cada um inclui um identificador com base em seu endereço MAC na mensagem de configuração, bem como o identificador da ponte que eles acreditam seja a raiz. Os endereços MAC são instalados pelo fabricante e garantidos para ser únicos em todo o mundo, o que torna esses identificadores convenientes e únicos. O bridges escolha a bridge com o identificador mais baixo para ser a raiz. Depois de bastante mensagens foram trocadas para espalhar a notícia, todas as pontes concordarão qual ponte é a raiz. Na Fig. 4-44, a ponte B_1 tem o identificador mais baixo e é vem a raiz.

Em seguida, é construída uma árvore de caminhos mais curtos da raiz a cada ponte. No

Fig. 4-44, as pontes B_2 e B_3 podem ser alcançadas cada uma da ponte B_1 diretamente, em um hop que é o caminho mais curto. A ponte B_4 pode ser alcançada em dois saltos, via B_2 ou B_3 . Para quebrar esse vínculo, o caminho através da ponte com o identificador mais baixo é escolhido,

então B_4 é alcançado por meio de B_2 . A ponte B_5 pode ser alcançada em dois saltos via B_3 . Para encontrar esses caminhos mais curtos, as pontes incluem a distância da raiz em seus mensagens de configuração. Cada ponte lembra o caminho mais curto que encontra para o raiz. As pontes então desligam as portas que não fazem parte do caminho mais curto.

Embora a árvore abranja todas as pontes, nem todos os links (ou mesmos pontes) são necessariamente presente na árvore. Isso acontece porque desligar as ameixas alguns links da rede para evitar loops. Mesmo depois que a árvore de abrangência foi estabelecido, o algoritmo continua a funcionar durante a operação normal para auto-detectar automaticamente as mudanças na topologia e atualizar a árvore.

O algoritmo para construir a árvore geradora foi inventado por Radia Perl-homem. Seu trabalho era resolver o problema de ingressar em LANs sem loops. Ela era teve uma semana para fazê-lo, mas ela teve a ideia para o algoritmo de árvore geradora ritmo em um dia. Felizmente, isso lhe deu tempo suficiente para escrevê-lo como um poema (Perl-homem, 1985):

*Acho que nunca verei
Um gráfico mais lindo do que uma árvore.
Uma árvore cuja propriedade crucial
É a conectividade sem loop.
Uma árvore que deve ter certeza de se estender.
Assim, os pacotes podem chegar a todas as LAN.
Primeiro, a raiz deve ser selecionada
Por ID é eleito.
Os caminhos de menor custo da raiz são traçados
Na árvore esses caminhos são colocados.
Uma malha é feita por pessoas como eu
Em seguida, as pontes encontram uma árvore geradora.*

Página 364

340

O SUBCAMADA DE CONTROLE DE ACESSO MÉDIO INDIVÍDUO. 4

O algoritmo de spanning tree foi então padronizado como IEEE 802.1D e usado para muitos anos. Em 2001, foi revisado para encontrar mais rapidamente uma nova árvore geradora após uma mudança de topologia. Para um tratamento detalhado das pontes, consulte Perlman (2000).

4.8.4 Repetidores, Hubs, Pontes, Switches, Roteadores e Gateways

Até agora neste livro, vimos uma variedade de maneiras de obter frames e pacotes de um computador para outro. Mencionamos repetidores, hubs, pontes, switches, roteadores e gateways. Todos esses dispositivos são de uso comum, mas todos eles diferem de maneiras sutis e não tão sutis. Uma vez que existem tantos eles, provavelmente vale a pena dar uma olhada neles juntos para ver o que laridades e diferenças são.

A chave para entender esses dispositivos é perceber que eles operam em diferentes camadas diferentes, conforme ilustrado na Figura 4-45 (a). A camada é importante porque diferentes os dispositivos usam diferentes informações para decidir como fazer a troca. Em um típico cenário, o usuário gera alguns dados para serem enviados a uma máquina remota. Esses dados são passados para a camada de transporte, que então adiciona um cabeçalho (por exemplo, um TCP cabeçalho) e passa a unidade resultante para a camada de rede. A rede camada adiciona seu próprio cabeçalho para formar um pacote de camada de rede (por exemplo, um pacote IP). No

Figura 4-45 (b), vemos o pacote IP sombreado em cinza. Em seguida, o pacote vai para o camada de enlace de dados, que adiciona seu próprio cabeçalho e soma de verificação (CRC) e dá a enviar quadro para a camada física para transmissão, por exemplo, em uma LAN.

Camada de aplicação

Gateway de aplicação
 Camada de transporte
 Portal de transporte
 Camada de rede
 Roteador
 Quadro, Armação
 cabeçalho
 Pacote
 cabeçalho
 TCP
 cabeçalho
 Pacote (fornecido pela camada de rede)
 Quadro (construído pela camada de link de dados)
 (b)
 (uma)
 Do utilizador
 dados
 CRC
 Camada de ligação de dados
 Bridge, switch
 Camada física
 Repetidor, hub

Figura 4-45. (a) Qual dispositivo está em qual camada. (b) Frames, pacotes e cabeçalhos.

Agora vamos dar uma olhada nos dispositivos de comutação e ver como eles se relacionam com o pacote

ets e frames. Na parte inferior, na camada física, encontramos os repetidores. Estes são dispositivos analógicos que funcionam com sinais nos cabos aos quais estão conectado. Um sinal que aparece em um cabo é limpo, amplificado e colocado em outro cabo. Os repetidores não entendem quadros, pacotes ou cabeçalhos. Eles desentender os símbolos que codificam bits como volts. Ethernet clássica, por exemplo, era

Página 365

SEC. 4,8

COMUTAÇÃO DA CAMADA DE LINK DE DADOS

341

projeto para permitir quatro repetidores que aumentariam o sinal para estender o máximo comprimento do cabo rum de 500 metros a 2500 metros.

Em seguida, chegamos aos hubs. Um hub tem várias linhas de entrada que ele une eletricamente. Os quadros que chegam em qualquer uma das linhas são enviados em todas as outras. E se

dois quadros chegam ao mesmo tempo, eles colidem, assim como em um cabo coaxial.

Todas as linhas que entram em um hub devem operar na mesma velocidade. Hubs diferem de repetidores em que eles (geralmente) não amplificam os sinais de entrada e são assinado para várias linhas de entrada, mas as diferenças são pequenas. Como repetidores, hubs são dispositivos da camada física que não examinam os endereços da camada de enlace ou os usam de qualquer forma.

Agora, vamos avançar para a camada de enlace de dados, onde encontramos pontes e es. Nós apenas estudamos as pontes com algum detalhe. Uma ponte conecta dois ou mais LANs. Como um hub, uma ponte moderna tem várias portas, geralmente o suficiente para 4 a 48 linhas de entrada de um certo tipo. Ao contrário de um hub, cada porta é isolada para ser sua domínio de colisão; se a porta tiver uma linha ponto-a-ponto full-duplex, o CSMA / CD algoritmo não é necessário. Quando um quadro chega, a ponte extrai o destino endereço do cabeçalho do quadro e procure em uma tabela para ver para onde enviar o quadro, Armação. Para Ethernet, este endereço é o endereço de destino de 48 bits mostrado em Fig. 4-14. A ponte só produz o quadro na porta onde é necessário e pode encaminhar vários quadros ao mesmo tempo.

As pontes oferecem um desempenho muito melhor do que os hubs, e o isolamento entre portas de ponte também significa que as linhas de entrada podem ser executadas em velocidades diferentes, possivelmente

mesmo com diferentes tipos de rede. Um exemplo comum é uma ponte com portas que conecte-se a Ethernet de 10, 100 e 1000 Mbps. O buffer dentro da ponte é precisava aceitar um quadro em uma porta e transmitir o quadro em um diferente porta. Se os quadros chegam mais rápido do que podem ser retransmitidos, a ponte pode funcionar fora do espaço do buffer e tem que começar a descartar quadros. Por exemplo, se um gigabit

Ethernet está despejando bits em uma Ethernet de 10 Mbps em alta velocidade, a ponte terá para armazená-los em buffer, na esperança de não ficar sem memória. Este problema ainda existe mesmo se

todas as portas funcionam na mesma velocidade porque mais de uma porta pode estar enviando quadros para uma determinada porta de destino.

As pontes foram originalmente concebidas para serem capazes de unir diferentes tipos de LANs, por exemplo, uma Ethernet e uma LAN Token Ring. No entanto, isso nunca funcionou bem por causa das diferenças entre as LANs. Diferentes formatos de quadro exigem copiar e reformatar, o que leva tempo de CPU, requer um novo cálculo de soma de verificação e introduz a possibilidade de erros não detectados devido a bits incorretos no memória da ponte. Diferentes comprimentos máximos de quadro também são um problema sério sem uma boa solução. Basicamente, os frames que são muito grandes para serem encaminhados devem

ser descartado. Tanto para transparência.

Duas outras áreas em que as LANs podem diferir são a segurança e a qualidade do serviço. Algumas LANs têm criptografia de camada de link, por exemplo 802.11, e outras não, para exemplo Ethernet. Algumas LANs têm recursos de qualidade de serviço, como prioridades, por exemplo 802.11, e alguns não, por exemplo Ethernet. Conseqüentemente, quando

Página 366

342

O SUBCAMADA DE CONTROLE DE ACESSO MÉDIO INDIVÍDUO. 4

um quadro deve viajar entre essas LANs, a segurança ou a qualidade do serviço esperada ed pelo remetente pode não ser capaz de ser fornecido. Por todas essas razões, o moderno pontes geralmente funcionam para um tipo de rede e roteadores, que veremos logo, são usados em vez de se juntar a redes de diferentes tipos.

Switches são pontes modernas com outro nome. As diferenças são mais para fazer mais com marketing do que questões técnicas, mas existem alguns pontos que vale a pena conhecer.

As pontes foram desenvolvidas quando a Ethernet clássica estava em uso, então elas tendem a se unir

tem poucas LANs e, portanto, relativamente poucas portas. O termo " mudar " é mais popular hoje em dia. Além disso, todas as instalações modernas usam links ponto a ponto, como cabos de par trançado, para que os computadores individuais se conectem diretamente a um switch e, assim,

o switch tenderá a ter muitas portas. Finalmente, " switch " também é usado como um gen- termo geral. Com uma ponte, a funcionalidade é clara. Por outro lado, um interruptor pode se referir a um switch Ethernet ou um tipo completamente diferente de dispositivo que toma decisões de encaminhamento, como uma comutação telefônica.

Até agora, vimos repetidores e hubs, que são bastante semelhantes, como bem como pontes e interruptores, que são ainda mais semelhantes entre si. Agora nós mude para roteadores, que são diferentes de todos os anteriores. Quando um pacote entra em um roteador, o cabeçalho do quadro e o trailer são retirados e o pacote citado no campo de carga útil do quadro (sombreado na Fig. 4-45) é passado para o roteamento Programas. Este software usa o cabeçalho do pacote para escolher uma linha de saída. Para um Pacote IP, o cabeçalho do pacote conterá um endereço de 32 bits (IPv4) ou 128 bits (IPv6), mas não um endereço IEEE 802 de 48 bits. O software de roteamento não vê o quadro endereços e nem mesmo sabe se o pacote veio em uma LAN ou linha ponto a ponto. Estudaremos roteadores e roteamento no capítulo. 5

Em outra camada, encontramos gateways de transporte. Estes conectam dois computadores que usam diferentes protocolos de transporte orientados à conexão. Por exemplo, suponha que um computador usando o protocolo TCP / IP orientado à conexão precisa se comunicar com um computador

computador usando um protocolo de transporte orientado a conexão diferente denominado SCTP. o gateway de transporte pode copiar os pacotes de uma conexão para outra, emaranhando-os conforme a necessidade.

Finalmente, os gateways de aplicativo entendem o formato e o conteúdo dos dados e pode traduzir mensagens de um formato para outro. Um gateway de e-mail poderia traduzir mensagens da Internet em mensagens SMS para telefones celulares, por exemplo. Como "switch," "gateway" é um termo genérico. Refere-se a um processo de proteção que funciona em uma camada alta.

4.8.5 LANs virtuais

Nos primeiros dias da rede local, grossos cabos amarelos serpenteavam os dutos de cabos de muitos prédios de escritórios. Cada computador que eles passaram era conectado. Nenhum pensamento foi dado a qual computador pertencia a qual LAN. Todas as pessoas nos escritórios adjacentes foram colocadas na mesma LAN, sejam elas desejados juntos ou não. A geografia superou os organogramas corporativos.

Página 367

SEC. 4,8

COMUTAÇÃO DA CAMADA DE LINK DE DADOS

343

Com o advento do par trançado e hubs na década de 1990, tudo mudou. Os edifícios foram religados (com despesas consideráveis) para arrancar todo o jardim amarelo mangueiras e instale pares trançados de cada escritório para armários de fiação centrais no final de cada corredor ou em uma sala de máquinas central, conforme ilustrado na Fig. 4-46. E se o vice-presidente encarregado da fiação era um visionário, pares trançados Categoria 5 foram instalados; se ele fosse um contador de feijão, o fio telefônico existente (categoria 3) foi usado (apenas para ser substituído alguns anos depois, quando surgiu a Ethernet rápida).

Par trançado
para um hub
Escritório
Interruptor
Cubo
Cubo
Corredor
Cabo
duto

Figura 4-46. Um prédio com fiação centralizada usando hubs e um switch.

Hoje, os cabos mudaram e os hubs se tornaram interruptores, mas os fios padrão de funcionamento ainda é o mesmo. Este padrão torna possível configurar LANs logicamente, em vez de fisicamente. Por exemplo, se uma empresa deseja k LANs, poderia comprar k switches. Escolhendo cuidadosamente quais conectores conectar em quais switches, os ocupantes de uma LAN podem ser escolhidos de uma forma que torne a organização sentido profissional, sem levar em conta a geografia.

Faz diferença quem está em qual LAN? Afinal, em quase todas as organizações, todas as LANs estão interconectadas. Resumindo, sim, muitas vezes é importante. Administração de rede

tratadores gostam de agrupar usuários em LANs para refletir a estrutura organizacional, em vez de do que o layout físico do edifício, por uma variedade de razões. Um problema é se-curiósidade. Uma LAN pode hospedar servidores da Web e outros computadores destinados ao público

usar. Outra LAN pode hospedar computadores contendo os registros do Human Re-departamento de fontes que não devem ser repassados para fora do departamento. Em um tal situação, colocando todos os computadores em uma única LAN e não deixando nenhum dos vers ser acessado de fora da LAN faz sentido. A gerência tende a franzir a testa ao ouvir que tal acordo é impossível.

Página 368

344

O SUBCAMADA DE CONTROLE DE ACESSO MÉDIO
INDIVÍDUO. 4

Um segundo problema é o carregamento. Algumas LANs são mais usadas do que outras e pode ser desejável separá-los. Por exemplo, se o pessoal da pesquisa for executado realizando todos os tipos de experimentos bacanas que às vezes saem do controle e saturam

sua LAN, o pessoal da administração pode não estar entusiasmado com a doação de alguns da capacidade que estavam usando para videoconferência para ajudar. Então de novo, isso pode impressionar a gerência com a necessidade de instalar uma rede mais rápida. Um terceiro problema é o tráfego de transmissão. As pontes transmitem o tráfego quando o local do destino é desconhecido e os protocolos da camada superior também usam transmissão. Por exemplo, quando um usuário deseja enviar um pacote para um endereço IP x , como isso sabe qual endereço MAC colocar no quadro? Vamos estudar esta questão em Indivíduo. 5, mas resumido brevemente, a resposta é que ele transmite um quadro de contendo a pergunta "quem possui o endereço IP x ?". Então ele espera por uma resposta. Como o número de computadores em uma LAN aumenta, assim como o número de broadcasts. Cada transmissão consome mais capacidade da LAN do que um quadro normal porque é entregue a todos os computadores da LAN. Ao manter as LANs não maiores do que eles precisa ser, o impacto do tráfego de transmissão é reduzido.

Relacionado às transmissões está o problema de que de vez em quando uma interface de rede vai quebrar ou ficar mal configurado e começar a gerar um fluxo infinito de quadros de transmissão. Se a rede for realmente azarada, alguns desses quadros provocarão respostas que levam a cada vez mais tráfego. O resultado desta **tempestade de transmissão** é que (1) toda a capacidade da LAN é ocupada por esses quadros e (2) todas as máquinas em todas as LANs interconectadas são paralisadas apenas processando e descartando todos os quadros sendo transmitidos.

A primeira vista, pode parecer que as tempestades de transmissão podem ser limitadas em escopo por separando as LANs com bridges ou switches, mas se o objetivo é conseguir transparency (ou seja, uma máquina pode ser movida para uma LAN diferente através da ponte com qualquer um que perceba), então as bridges precisam encaminhar quadros de broadcast. Tendo visto por que as empresas podem querer várias LANs com restrições escopos, vamos voltar ao problema de desacoplar a topologia lógica da topologia física. Construir uma topologia física para refletir a estrutura organizacional. A estrutura pode agregar trabalho e custo, mesmo com cabeamento e interruptores centralizados. Para por exemplo, se duas pessoas no mesmo departamento trabalham em edifícios diferentes, pode ser mais fácil conectá-las a switches diferentes que fazem parte de LANs diferentes. Até se este não for o caso, um usuário pode ser transferido dentro da empresa de um departamento para outro sem mudar de escritório, ou pode mudar de escritório sem mudar departamentos. Isso pode resultar no usuário estar na LAN errada até que um administrador altera o conector do usuário de um switch para outro. Além disso, o número de computadores que pertencem a diferentes departamentos não pode ser uma boa combinação para o número de portas nos switches; alguns departamentos podem ser muito pequenos e outros tão grandes que requerem vários interruptores. Isto resulta em portas de switch perdidas que não são usadas.

Em muitas empresas, as mudanças organizacionais ocorrem o tempo todo, o que significa que administradores de sistema passam muito tempo retirando os plugues e empurrando-os de volta

Página 369

SEC. 4,8

COMUTAÇÃO DA CAMADA DE LINK DE DADOS

345

em outro lugar. Além disso, em alguns casos, a mudança não pode ser feita de todo sem fazer com que o par trançado da máquina do usuário esteja muito longe da chave correta (por exemplo, no prédio errado) ou as portas de switch disponíveis estão na LAN errada. Em resposta às solicitações dos clientes por mais flexibilidade, os fornecedores de rede começaram trabalhando em uma maneira de religar edifícios inteiramente em software. O conceito resultante é chamado de **VLAN** (**Virtual LAN**). Ele foi padronizado pelo IEEE 802 comitê e agora está amplamente implantado em muitas organizações. Vamos agora dar uma olhe para isso. Para obter informações adicionais sobre VLANs, consulte Seifert e Edwards (2008).

As VLANs são baseadas em switches com reconhecimento de VLAN. Para configurar uma rede baseada em VLAN-

funcionar, o administrador da rede decide quantas VLANs haverá, quais os computadores estarão em qual VLAN e como as VLANs serão chamadas. Frequentemente as VLANs são (informalmente) nomeadas por cores, já que então é possível imprimir diagramas de cores mostrando o layout físico das máquinas, com os membros de a LAN vermelha em vermelho, os membros da LAN verde em verde e assim por diante. Nesse caminho,

os layouts físicos e lógicos são visíveis em uma única visualização.

Como exemplo, considere a LAN com ponte da Figura 4-47, na qual nove das

as máquinas pertencem à VLAN G (cinza) e cinco pertencem à VLAN W (branca).

As máquinas da VLAN cinza estão espalhadas por dois switches, incluindo dois m-

achines que se conectam a um switch por meio de um hub.

Estação cinza

B1

B2

Cubo

G

W

GW

W

G

G

G

GW

G

G

G

W

G

W

Estação branca

Porto cinza

Porto branco

Cinza e

Porto branco

Ponte

Figura 4-47. Duas VLANs, cinza e branca, em uma LAN com ponte.

Para que as VLANs funcionem corretamente, as tabelas de configuração devem ser definidas nas pontes. Essas tabelas informam quais VLANs são acessíveis por meio de quais portas.

Quando um quadro vem, digamos, da VLAN cinza, ele deve ser encaminhado para todos as portas marcadas com um G. Isso vale para o tráfego comum (ou seja, unicast) para o qual as pontes não aprenderam a localização do destino, bem como para transmitir e transmitir tráfego. Observe que uma porta pode ser rotulada com várias VLAN cores.

Por exemplo, suponha que uma das estações cinza conectada à ponte B1 em

A Figura 4-47 envia um quadro a um destino que não foi observado de antemão.

A ponte B1 receberá o quadro e verá que ele veio de uma máquina no cinza

VLAN, por isso vai inundar o quadro em todas as portas rotuladas G (exceto o de entrada porta). O quadro será enviado para as outras cinco estações cinza anexadas ao B1 também como no link de B1 para a ponte B2 . Na ponte B2 , o quadro é similarmente para- protegido em todas as portas rotuladas G. Isso envia o quadro para uma outra estação e o hub (que transmitirá o quadro para todas as suas estações). O hub tem ambos os rótulos porque ele se conecta a máquinas de ambas as VLANs. O quadro não é enviado em outras portas sem G no rótulo porque a ponte sabe que não há m-

achines na VLAN cinza que pode ser alcançada por meio dessas portas.

Em nosso exemplo, o quadro é enviado apenas da ponte B1 para a ponte B2 porque existem máquinas na VLAN cinza que estão conectadas a B2 . Olhando para o VLAN branca, podemos ver que a porta B2 da ponte que se conecta à ponte B1 não é rotulado como W. Isso significa que um quadro na VLAN branca não será encaminhado

da ponte *B2* para a ponte *B1*. Este comportamento está correto porque nenhuma estação no VLAN brancas estão conectadas a *B1*.

O padrão IEEE 802.1Q

Para implementar este esquema, as pontes precisam saber em qual VLAN um quadro original pertence. Sem esta informação, por exemplo, quando a ponte *B2* obtém um quadro da ponte *B1* na Fig. 4-47, ele não pode saber se deve encaminhar o quadro na VLAN cinza ou branca. Se estivéssemos projetando um novo tipo de LAN, seja fácil adicionar apenas um campo VLAN no cabeçalho. Mas o que fazer sobre Ethernet, que é a LAN dominante, e não tinha campos sobressalentes em torno do identificador de VLAN?

O comitê IEEE 802 teve esse problema lançado em seu colo em 1995. Após muita discussão, fez o impensável e mudou o cabeçalho Ethernet. o novo formato foi publicado no padrão IEEE **802.1Q**, emitido em 1998. O novo formato contém uma tag VLAN; vamos examiná-lo em breve. Não surpreendentemente, chang-criar algo tão bem estabelecido quanto o cabeçalho Ethernet não era totalmente trivial.

Algumas perguntas que vêm à mente são:

1. Precisamos jogar fora várias centenas de milhões de placas Ethernet existentes?
2. Se não, quem gera os novos campos?
3. O que acontece com os quadros que já têm o tamanho máximo?

Claro, o comitê 802 estava (muito dolorosamente) ciente desses problemas e teve que encontrar soluções, o que aconteceu.

A chave para a solução é perceber que os campos VLAN são apenas realmente usado pelas pontes e switches e *não* pelas máquinas dos usuários. Assim, na Fig. 4-47, não é realmente essencial que eles estejam presentes nas linhas que vão até o final estações, desde que estejam na linha entre as pontes. Além disso, para usar VLANs, as pontes devem estar cientes de VLAN. Esse fato torna o projeto viável.

Página 371

SEC. 4,8

COMUTAÇÃO DA CAMADA DE LINK DE DADOS

347

Quanto a descartar todas as placas Ethernet existentes, a resposta é não. Lembrar que o comitê 802.3 não conseguia nem mesmo fazer com que as pessoas mudassem o campo *Tipo* para um

Campo de *comprimento*. Você pode imaginar a reação a um anúncio de que todos os existentes As placas Ethernet tiveram que ser descartadas. No entanto, as novas placas Ethernet são 802.1Q compatível e pode preencher corretamente os campos da VLAN.

Como pode haver computadores (e switches) que não reconhecem VLAN, o a primeira ponte com reconhecimento de VLAN a tocar um quadro adiciona campos de VLAN e o último

na estrada os remove. Um exemplo de uma topologia mista é mostrado em Fig. 4-48. Nesta figura, os computadores com reconhecimento de VLAN geram marcados (ou seja, 802.1Q)

quadros diretamente, e comutação posterior usa essas tags. Os símbolos sombreados são VLAN-cientes e os vazios não.

Legado

ponte

e hospedeiro

B1

B2

B5

Marcado

quadro, Armação

B4

B3

B6

Ciente de VLAN

hospedeiro e ponte

Legado

quadro, Armação

Figura 4-48. LAN com ponte que reconhece apenas parcialmente a VLAN. O símbolo sombreado

ols estão cientes de VLAN. Os vazios não são.

Com 802.1Q, os quadros são coloridos dependendo da porta na qual eles são received. Para que este método funcione, todas as máquinas em uma porta devem pertencer ao mesmo VLAN, o que reduz a flexibilidade. Por exemplo, na Figura 4-48, esta propriedade é válida para todas as portas onde um computador individual se conecta a uma ponte, mas não para o porta onde o hub se conecta à ponte *B2*.

Além disso, a ponte pode usar o protocolo de camada superior para selecionar a cor.

Desta forma, os quadros que chegam em uma porta podem ser colocados em diferentes VLANs dependente se eles carregam pacotes IP ou quadros PPP.

Outros métodos são possíveis, mas não são suportados pelo 802.1Q. Como um exemplo, o endereço MAC pode ser usado para selecionar a cor da VLAN. Isso pode ser útil para frames vindos de uma LAN 802.11 próxima na qual os laptops enviam quadros por meio de portas diferentes à medida que se movem. Um endereço MAC seria então mapeado

para uma VLAN fixa, independentemente da porta em que ela entrou na LAN.

Quanto ao problema de quadros com mais de 1518 bytes, 802.1Q apenas levantou o limite de 1522 bytes. Felizmente, apenas computadores e switches compatíveis com VLAN devem apoiar esses quadros mais longos.

Agora, vamos dar uma olhada no formato de quadro 802.1Q. Isso é mostrado na Fig. 4-49.

A única mudança é a adição de um par de campos de 2 bytes. O primeiro é o

Página 372

348

O SUBCAMADA DE CONTROLE DE ACESSO MÉDIO INDIVÍDUO. 4

ID do protocolo VLAN. Ele sempre tem o valor 0x8100. Como este número é maior do que 1500, todas as placas Ethernet interpretam-no como um tipo e não como um comprimento. Que

cartão legado faz com tal quadro é discutível, uma vez que esses quadros não deveriam ser enviado para cartões legados.

802.3
comprimento
Dados
Almofada
Verifica-
soma
Destino
endereço
Fonte
endereço
802.1Q
comprimento
Dados
Almofada
Tag
Identificador VLAN
Protocolo VLAN
ID (0x8100)
Pri
C
F
Eu
Verifica-
soma
Destino
endereço
Fonte
endereço

Figura 4-49. Os formatos de quadro Ethernet 802.3 (legado) e 802.1Q.

O segundo campo de 2 bytes contém três subcampos. O principal é a *VLAN identificador*, ocupando os 12 bits de ordem inferior. Isso é o que a coisa toda é sobre - a cor da VLAN à qual o quadro pertence. A *prioridade de 3 bits* campo não tem nada a ver com VLANs, mas desde a alteração do cabeçalho Ethernet é um evento que ocorre uma vez em uma década, que leva três anos e apresenta uma centena de pessoas, por que não colocar outras coisas boas enquanto você está nisso? Este campo torna possível distinguir o tráfego pesado em tempo real do tráfego moderado em tempo real do tempo

tráfego insensível para fornecer melhor qualidade de serviço sobre Ethernet. Isto é necessário para voz sobre Ethernet (embora com toda a justiça, IP teve um campo semelhante por um quarto de século e ninguém nunca o usou).

O último campo, *CFI* (*indicador de formato canônico*), deveria ter sido chamado de *CEI* (*indicador de ego corporativo*). A intenção original era indicar a ordem de os bits nos endereços MAC (little-endian versus big-endian), mas esse uso conseguiu perdido em outras controvérsias. Sua presença agora indica que a carga útil contém um quadro 802.5 liofilizado que espera encontrar outra LAN 802.5 no destino enquanto é transportado por Ethernet no meio. Todo esse arranjo, de claro, não tem nada a ver com VLANs. Mas o comitê de padrões a política não é diferente da política normal: se você votar na minha parte, votarei na sua mordeu.

Como mencionamos acima, quando um frame marcado chega a um VLAN-aware switch, o switch usa o identificador VLAN como um índice em uma tabela para descobrir em quais portas enviar. Mas de onde vem a mesa? Se for manualmente construída, estamos de volta à estaca zero: configuração manual de pontes. A beleza da ponte transparente é que ela é plug-and-play e não requer nenhuma configuração manual. Seria uma pena perder essa propriedade. Para-felizmente, as pontes com reconhecimento de VLAN também podem se autoconfigurar com base em observando as tags que aparecem. Se um quadro marcado como VLAN 4 entrar na porta

Página 373

SEC. 4,8

COMUTAÇÃO DA CAMADA DE LINK DE DADOS

349

3, aparentemente, alguma máquina na porta 3 está na VLAN 4. O padrão 802.1Q ex-planeja como construir as tabelas dinamicamente, principalmente referenciando por-do padrão 802.1D.

Antes de deixar o assunto de roteamento VLAN, vale a pena fazer um último observação. Muitas pessoas nos mundos da Internet e Ethernet são fanaticamente a favor de redes sem conexão e se opõe violentamente a qualquer coisa que chegue de conexões no link de dados ou camadas de rede. No entanto, as VLANs apresentam algumas coisas que são surpreendentemente semelhantes a uma conexão. Para usar VLANs corretamente, cada frame carrega um novo identificador especial que é usado como um índice em uma tabela dentro do switch para verificar para onde o quadro deve ser enviado. Isso é precisamente o que acontece em redes orientadas a conexão. Em redes sem conexão, é o endereço de destino que é usado para roteamento, não algum tipo de identificação de conexão. Veremos mais sobre esse crescente conexionalismo no cap. 5

4.9 RESUMO

Algumas redes possuem um único canal que é usado para todas as comunicações. No essas redes, a questão-chave do projeto é a alocação deste canal entre as estações concorrentes que desejam usá-lo. FDM e TDM são aloca-esquemas de operação quando o número de estações é pequeno e fixo e o tráfego é contínuo. Ambos são amplamente usados nessas circunstâncias, por exemplo, para dividir aumentando a largura de banda em troncos de telefone. No entanto, quando o número de estações é grande e variável ou o tráfego é bastante intermitente - o caso comum em computadores redes — FDM e TDM são escolhas ruins.

Numerosos algoritmos de alocação dinâmica de canais foram desenvolvidos. O protocolo ALOHA, com e sem slotting, é usado em muitos derivados reais sistemas, por exemplo, modems a cabo e RFID. Como uma melhoria quando o o estado do canal pode ser detectado, as estações podem evitar iniciar uma transmissão enquanto outra estação está transmitindo. Esta técnica, detecção de portador, levou a uma variedade de protocolos CSMA para LANs e MANs. É a base para Ethernet clássica e Redes 802.11.

Uma classe de protocolos que elimina totalmente a contenção ou, pelo menos, a reduz consideravelmente, é bem conhecido. O protocolo de bitmap, topologias como anéis e

o protocolo binário de contagem regressiva elimina completamente a contenção. A caminhada da árvore

protocolo o reduz, dividindo dinamicamente as estações em dois grupos separados de tamanhos diferentes e permitindo contenção apenas dentro de um grupo; idealmente aquele grupo é escolhido de forma que apenas uma estação esteja pronta para enviar quando for permitido. As LANs sem fio têm os problemas adicionais de que é difícil sentir a colisão transmissões e que as regiões de cobertura das estações podem ser diferentes. No dom-LAN wireless inant, IEEE 802.11, estações usam CSMA / CA para mitigar o primeiro problema, deixando pequenas lacunas para evitar colisões. As estações também podem usar o Protocolo RTS / CTS para combater terminais ocultos que surgem por causa do segundo

Página 374

350

O SUBCAMADA DE CONTROLE DE ACESSO MÉDIO INDIVÍDUO. 4

problema. IEEE 802.11 é comumente usado para conectar laptops e outros dispositivos a pontos de acesso sem fio, mas também pode ser usado entre dispositivos. Qualquer um de vários camadas físicas podem ser usadas, incluindo FDM multicanal com e sem multi antenas ple e espectro de propagação.

Como 802.11, leitores e tags RFID usam um protocolo de acesso aleatório para comunicar identificadores icate. Outros PANs e MANs sem fio têm designs diferentes. o O sistema Bluetooth conecta fones de ouvido e muitos tipos de periféricos a computadores sem fios. IEEE 802.16 fornece um serviço de dados de Internet sem fio de área ampla para computadores fixos e móveis. Ambas as redes usam um sistema centralizado, design orientado a conexão em que o mestre Bluetooth e a base WiMAX estação decide quando cada estação pode enviar ou receber dados. Para 802.16, este design suporta qualidade de serviço diferente para tráfego em tempo real, como chamadas telefônicas e tráfego interativo, como navegação na web. Para Bluetooth, colocando a complexidade em o mestre leva a dispositivos escravos baratos.

Ethernet é a forma dominante de LAN com fio. Ethernet clássica usada CSMA / CD para alocação de canal em um cabo amarelo do tamanho de uma mangueira de jardim que

serpenteava de máquina em máquina. A arquitetura mudou conforme as velocidades aumentou de 10 Mbps para 10 Gbps e continua a subir. Agora, links ponto a ponto como o par trançado são anexados a hubs e switches. Com interruptores modernos e links full-duplex, não há contenção nos links e o switch pode encaminhar quadros entre portas diferentes em paralelo.

Com edifícios cheios de LANs, é necessária uma maneira de interconectar todos eles. Plugue-pontes and-play são usadas para esse fim. As pontes são construídas com uma estrutura algoritmo de aprendizagem e um algoritmo de árvore geradora. Uma vez que esta funcionalidade é construída

em switches modernos, os termos " bridge " e " switch " são usados alternadamente.

Para ajudar no gerenciamento de LANs com ponte, as VLANs permitem que a topologia física ser dividido em diferentes topologias lógicas. O padrão VLAN, IEEE 802.1Q, apresenta um novo formato para frames Ethernet.

PROBLEMAS

1. Para este problema, use uma fórmula deste capítulo, mas primeiro declare a fórmula. Molduras chegam aleatoriamente em um canal de 100 Mbps para transmissão. Se o canal estiver ocupado quando um quadro chega, ele espera sua vez em uma fila. O comprimento do quadro é distribuído exponencialmente com uma média de 10.000 bits / quadro. Para cada uma das seguintes taxas de chegada de quadro, dê o atraso experimentado pelo quadro médio, incluindo o tempo de fila e tempo de missão.

- (a) 90 quadros / seg.
- (b) 900 quadros / seg.
- (c) 9000 quadros / seg.

Página 375

INDIVÍDUO. 4

PROBLEMAS

351

2. Um grupo de N estações compartilha um canal ALOHA puro de 56 kbps. Cada estação produz um Quadro de 1000 bits em média uma vez a cada 100 segundos, mesmo se o anterior ainda não enviado (por exemplo, as estações podem armazenar em buffer frames de saída). Qual é o valor máximo de N ?
3. Considere o atraso de ALOHA puro versus ALOHA com fenda em carga baixa. Qual deles é Menos? Explique sua resposta.
4. Uma grande população de usuários ALOHA consegue gerar 50 solicitações / s, incluindo originais e retransmissões. O tempo é dividido em unidades de 40 mseg.
- Qual é a chance de sucesso na primeira tentativa?
 - Qual é a probabilidade de exatamente k colisões e então um sucesso?
 - Qual é o número esperado de tentativas de transmissão necessárias?
5. Em um sistema ALOHA de população infinita com slots, o número médio de slots de uma estação espera entre uma colisão e uma retransmissão é 4. Trace o atraso em relação ao rendimento curva para este sistema.
6. Qual é o comprimento de um slot de contenção em CSMA / CD para (a) um cabo condutor duplo de 2 km (a velocidade de propagação do sinal é 82% da velocidade de propagação do sinal no vácuo)? e (b) um cabo de fibra óptica multimodo de 40 km (a velocidade de propagação do sinal é de 65% da velocidade de propagação no vácuo)?
7. Quanto tempo uma estação, s , tem que esperar no pior dos casos antes de começar a transmitir seu frame em uma LAN que usa o protocolo básico de bitmap?
8. No protocolo binário de contagem regressiva, explique como uma estação de número inferior pode ser morrendo de fome de enviar um pacote.
9. Dezesseis estações, numeradas de 1 a 16, estão disputando o uso de um canal compartilhado nel usando o protocolo adaptativo de caminhada em árvore. Se todas as estações cujos endereços são números primos de repente ficam prontos de uma vez, quantos slots de bits são necessários para resolver a contenção?
10. Considere cinco estações sem fios, U_m, B, C, D , e E . A estação A pode se comunicar com todos outras estações. B pode se comunicar com um, C e E . C pode se comunicar com A, B e D . D pode comunicar com um, C e E . E pode comunicar A, D e B .
- Quando A está enviando para B , que outras comunicações são possíveis?
 - Quando B está enviando para A , que outras comunicações são possíveis?
 - Quando B está enviando para C , que outras comunicações são possíveis?
11. Seis estações, de A a F , se comunicam usando o protocolo MACA. É possível para duas transmissões ocorrer simultaneamente? Explique sua resposta.
12. Um prédio de escritórios de sete andares tem 15 escritórios adjacentes por andar. Cada escritório contém um tomada de parede para um terminal na parede frontal, para que as tomadas formem uma grade retangular em o plano vertical, com separação de 4 m entre as tomadas, tanto na horizontal como verticalmente. Supondo que seja possível passar um cabo reto entre qualquer par de tomadas, horizontalmente, verticalmente ou diagonalmente, quantos metros de cabo são necessários para conectar todos os soquetes usando
- Uma configuração em estrela com um único roteador no meio?
 - Uma LAN 802.3 clássica?

Página 376

352

O SUBCAMADA DE CONTROLE DE ACESSO MÉDIO

INDIVÍDUO. 4

13. Qual é a taxa de transmissão da Ethernet clássica de 10 Mbps?
14. Esboce a codificação Manchester em uma Ethernet clássica para o fluxo de bits 0001110101.
15. Uma LAN CSMA / CD de 10 Mbps com 1 km de comprimento (não 802.3) tem uma velocidade de propagação de 200 m / μ sec. Repetidores não são permitidos neste sistema. Os quadros de dados têm 256 bits, incluindo 32 bits de cabeçalho, soma de verificação e outras sobrecargas. O primeiro slot de bit após um a transmissão bem-sucedida é reservada para o receptor capturar o canal, a fim de enviar um quadro de confirmação de 32 bits. Qual é a taxa de dados efetiva, excluindo sobrecarga, supondo que não haja colisões?
16. Duas estações CSMA / CD estão tentando transmitir arquivos longos (multiframe). Depois de cada quadro é enviado, eles disputam o canal, usando o backoff exponencial binário algoritmo. Qual é a probabilidade de que a contenção termine na rodada k , e qual é a número médio de rodadas por período de contenção?
17. Um pacote IP a ser transmitido por Ethernet tem 60 bytes de comprimento, incluindo todos os seus cabeçalhos. Se LLC não estiver em uso, é necessário preenchimento no quadro Ethernet e, em caso afirmativo, quantos bytes?

- 18.** Os frames Ethernet devem ter pelo menos 64 bytes de comprimento para garantir que o transmissor ainda esteja indo em caso de uma colisão na outra extremidade do cabo. Fast Ethernet tem o mesmo Tamanho mínimo de quadro de 64 bytes, mas pode obter os bits dez vezes mais rápido. Como é possível manter o mesmo tamanho mínimo de quadro?
- 19.** Alguns livros citam o tamanho máximo de um quadro Ethernet como 1522 bytes em vez de 1500 bytes. Eles estão errados? Explique sua resposta.
- 20.** Quantos quadros por segundo a Ethernet gigabit consegue lidar? Pense bem e pegue em consideração todos os casos relevantes. Dica : o fato de ser uma Ethernet *gigabit* é importante.
- 21.** Cite duas redes que permitem que os quadros sejam empacotados consecutivamente. Por que esse recurso Vale a pena ter?
- 22.** Na Fig. 4-27, quatro estações, *A* , *B* , *C* e *D* , são mostradas. Qual das duas últimas estações você acha que está mais próximo de *A* e por quê?
- 23.** Dê um exemplo para mostrar que o RTS / CTS no protocolo 802.11 é um pouco diferente do que no protocolo MACA.
- 24.** Uma LAN sem fio com um AP possui 10 estações clientes. Quatro estações têm taxas de dados de 6 Mbps, quatro estações têm taxas de dados de 18 Mbps e as duas últimas estações têm dados taxas de 54 Mbps. Qual é a taxa de dados experimentada por cada estação quando todas as dez estações estão enviando dados juntos, e
- (a) TXOP não é usado?
 - (b) TXOP é usado?
- 25.** Suponha que uma LAN 802.11b de 11 Mbps está transmitindo quadros de 64 bytes consecutivos em um canal de rádio com uma taxa de erro de bits de 10⁻⁷. Quantos quadros por segundo vão ser danificado em média?
- 26.** Uma rede 802.16 tem uma largura de canal de 20 MHz. Quantos bits / s podem ser enviados para uma estação de assinante?

Página 377

INDIVÍDUO. 4 PROBLEMAS

353

- 27.** Cite duas razões pelas quais as redes podem usar um código de correção de erros em vez de um detecção e retransmissão.
- 28.** Liste duas maneiras em que o WiMAX é semelhante ao 802.11 e duas maneiras em que é diferente de 802.11.
- 29.** Na Fig. 4-34, vemos que um dispositivo Bluetooth pode estar em duas piconets ao mesmo tempo. Existe alguma razão pela qual um dispositivo não pode ser o mestre em ambos no mesmo tempo?
- 30.** Qual é o tamanho máximo do campo de dados para um quadro Bluetooth de 3 slots na taxa básica? Explique sua resposta.
- 31.** A Figura 4-24 mostra vários protocolos da camada física. Qual destes está mais próximo do Protocolo de camada física Bluetooth? Qual é a maior diferença entre os dois?
- 32.** É mencionado na Seção 4.6.6 que a eficiência de um quadro de 1 slot com encodificação é de cerca de 13% na taxa de dados básica. Qual será a eficiência se um quadro de 5 slots com a codificação de repetição é usada na taxa de dados básica em vez disso?
- 33.** Os quadros de beacon na variante de espectro de propagação de salto de frequência de 802.11 contêm o tempo de permanência. Você acha que os quadros de beacon análogos no Bluetooth também contêm o tempo de permanência? Discuta sua resposta.
- 34.** Suponha que haja 10 tags RFID ao redor de um leitor RFID. Qual é o melhor valor de Q? Qual é a probabilidade de uma tag responder sem colisão em um determinado slot?
- 35.** Liste algumas das preocupações de segurança de um sistema RFID.
- 36.** Um switch projetado para uso com Ethernet rápida tem um painel traseiro que pode mover 10 Gbps. Quantos quadros / s ele pode lidar no pior caso?
- 37.** Descreva resumidamente a diferença entre os switches store-and-forward e cut-through.
- 38.** Considere a LAN estendida conectada usando as pontes *B1* e *B2* na Figura 4-41 (b). Supõe as tabelas de hash nas duas pontes estão vazias. Liste todas as portas nas quais um pacote irá ser encaminhado para a seguinte sequência de transmissões de dados:
- (a) *Um* envia um pacote para *C* .
 - (b) *E* envia um pacote para *F* .
 - (c) *F* envia um pacote para *E* .
 - (d) *L* envia um pacote para *E* .
 - (e) *D* envia um pacote para *um* .
 - (f) *B* envia um pacote para *F* .
- 39.** Os interruptores armazenar e encaminhar têm uma vantagem sobre os interruptores cut-through com respeito para quadros danificados. Explique o que é.

- 40.** É mencionado na Seção 4.8.3 que algumas pontes podem nem mesmo estar presentes no vazio árvore nenhuma. Descreva um cenário em que uma ponte pode não estar presente na árvore geradora.
41. Para fazer as VLANs funcionarem, são necessárias tabelas de configuração nas pontes. E se o As VLANs da Figura 4-47 usavam hubs em vez de switches? Os hubs precisam de configuração mesmas também? Por que ou por que não?

Página 378

354

O SUBCAMADA DE CONTROLE DE ACESSO MÉDIO

INDIVÍDUO. 4

- 42.** Na Figura 4-48, o switch no domínio final legado à direita é um VLAN-aware interruptor. Seria possível usar um switch legado lá? Se sim, como isso trabalhos? Se não, porque não?
43. Escreva um programa para simular o comportamento do protocolo CSMA / CD sobre Ethernet quando há N estações prontas para transmitir enquanto um quadro está sendo transmitido. Seu programa deve relatar os horários em que cada estação começa a enviar com sucesso quadro. Suponha que um tique do relógio ocorre uma vez a cada slot de tempo (51,2 µseg) e um a detecção de íons e o envio de uma sequência de interferência leva um slot de tempo. Todos os quadros são o comprimento máximo permitido.

Página 379

5

A CAMADA DE REDE

A camada de rede está preocupada em obter pacotes da origem de todos os caminhos para o destino. Chegar ao destino pode exigir muitos saltos em roteadores intermediários ao longo do caminho. Esta função contrasta claramente com a de a camada de enlace de dados, que tem o objetivo mais modesto de apenas mover quadros de uma extremidade de um fio para a outra. Assim, a camada de rede é a camada mais baixa que lida com transmissão ponta a ponta.

Para atingir seus objetivos, a camada de rede deve saber sobre a topologia do rede (ou seja, o conjunto de todos os roteadores e links) e escolher os caminhos apropriados por meio mesmo para grandes redes. Também deve ter cuidado ao escolher as rotas para evitar sobrecarregando algumas das linhas de comunicação e roteadores, deixando outros ocioso. Finalmente, quando a origem e o destino estão em redes diferentes, novos problemas ocorrem. Cabe à camada de rede lidar com eles. Neste capítulo vamos estudar todas essas questões e ilustrá-las, principalmente usando a Internet e seu protocolo de camada de rede, IP.

5.1 PROBLEMAS DE DESIGN DA CAMADA DE REDE

Nas seções a seguir, daremos uma introdução a alguns dos problemas que os projetistas da camada de rede devem enfrentar. Esses problemas incluem o serviço prestado à camada de transporte e o desenho interno da rede.

355

Página 380

356

A CAMADA DE REDE

INDIVÍDUO. 5

5.1.1 Comutação de pacotes de armazenamento e encaminhamento

Antes de começar a explicar os detalhes da camada de rede, vale a pena reafirmar o contexto no qual os protocolos da camada de rede operam. Este contexto pode ser

visto na Fig. 5-1. Os principais componentes da rede são os equipamentos do ISP (roteadores conectados por linhas de transmissão), mostrado dentro do oval sombreado, e o equipamento dos clientes, mostrado fora do oval. Host *H1* está diretamente conectado a um dos roteadores do ISP, *A*, talvez como um computador doméstico que está conectado a um Modem DSL. Em contraste, *H2* está em uma LAN, que pode ser uma Ethernet de escritório, com um roteador, *F*, de propriedade e operado pelo cliente. Este roteador tem um aluguel linha para o equipamento do ISP. Mostramos *F* como estando fora do oval porque não pertence ao ISP. Para os fins deste capítulo, no entanto, roteadores nas instalações do cliente são consideradas parte da rede ISP porque executam os mesmos algoritmos que os roteadores do ISP (e nossa principal preocupação aqui são os algoritmos).

```

D
C
B
UMA
E
F
Pacote
Processo P1
Host H1
Roteador
Equipamento do ISP
H2
LAN
P2

```

Figura 5-1. O ambiente dos protocolos da camada de rede.

Este equipamento é usado da seguinte maneira. Um host com um pacote para enviar o transmite para o roteador mais próximo, em sua própria LAN ou por meio de um link ponto a ponto com o ISP. O pacote é armazenado lá até que chegue totalmente e o link termine de processamento verificando a soma de verificação. Em seguida, ele é encaminhado para o próximo roteador ao longo do caminho até chegar ao host de destino, onde é entregue. Isto mecanismo é a comutação de pacotes armazenar e encaminhar, como vimos anteriormente capítulos.

5.1.2 Serviços prestados à camada de transporte

A camada de rede fornece serviços para a camada de transporte na rede camada / interface da camada de transporte. Uma questão importante é precisamente que tipo de serviços que a camada de rede fornece à camada de transporte. Os serviços precisam ser cuidadosamente projetado com os seguintes objetivos em mente:

Página 381

SEC. 5,1

QUESTÕES DE DESIGN DA CAMADA DE REDE

357

1. Os serviços devem ser independentes da tecnologia do roteador.

2. A camada de transporte deve ser protegida do número, tipo e topologia dos roteadores presentes.

3. Os endereços de rede disponibilizados para a camada de transporte devem usar um plano de numeração uniforme, mesmo em LANs e WANs.

Dados esses objetivos, os designers da camada de rede têm muita liberdade em escrever especificações detalhadas dos serviços a serem oferecidos à camada de transporte. Essa liberdade geralmente degenera em uma batalha violenta entre duas facções em conflito.

A discussão se concentra em se a camada de rede deve fornecer conexão serviço orientado ou serviço sem conexão.

Um campo (representado pela comunidade da Internet) argumenta que os roteadores trabalho é mover pacotes e nada mais. Nesta visão (com base em 40 anos de experiência com uma rede de computador real), a rede é inherentemente não confiável, não importa como é projetado. Portanto, os anfitriões devem aceitar este fato e fazer controle de erros (ou seja, detecção e correção de erros) e o próprio controle de fluxo.

Este ponto de vista leva à conclusão de que o serviço de rede deve ser connectionless, com primitivos SEND PACKET e RECEIVE PACKET e pouco mais.

Em particular, nenhum pedido de pacote e controle de fluxo deve ser feito, porque os anfitriões farão isso de qualquer maneira e geralmente há pouco a ganhar fazendo

duas vezes. Este raciocínio é um exemplo do **argumento ponta a ponta**, um design princípio que tem sido muito influente na formação da Internet (Saltzer et al., 1984). Além disso, cada pacote deve levar o endereço de destino completo, porque cada pacote enviado é transportado independentemente de seus predecessores, se houver. O outro campo (representado pelas companhias telefônicas) argumenta que a rede trabalho deve fornecer um serviço orientado para conexão confiável. Eles afirmam que 100 anos de experiência de sucesso com o sistema telefônico mundial é um excelente guia emprestado. Nesta visão, a qualidade do serviço é o fator dominante, e sem conexões na rede, a qualidade do serviço é muito difícil de alcançar, especialmente especialmente para tráfego em tempo real, como voz e vídeo.

Mesmo depois de várias décadas, essa controvérsia ainda está muito viva. Cedo, redes de dados amplamente utilizadas, como o X.25 na década de 1970 e seu sucessor Frame Relay na década de 1980, eram orientados à conexão. No entanto, desde os dias do ARPANET e o início da Internet, as camadas de rede sem conexão cresceram tremendamente em popularidade. O protocolo IP é agora um símbolo sempre presente de sucesso. Não foi intimidado por uma tecnologia orientada a conexão chamada ATM que foi desenvolvida para derrubá-lo na década de 1980; em vez disso, é ATM que agora se encontra em nichos de uso e IP que está assumindo as redes telefônicas. Nos bastidores, como sempre, a Internet está desenvolvendo recursos orientados à conexão como qualidade de serviço, vem mais importante. Dois exemplos de tecnologias orientadas à conexão são MPLS (MultiProtocol Label Switching), que descreveremos neste capítulo, e VLANs, que vimos no cap. 4. Ambas as tecnologias são amplamente utilizadas.

Página 382

358

A CAMADA DE REDE
INDIVÍDUO. 5

5.1.3 Implementação do serviço sem conexão

Tendo olhado para as duas classes de serviço que a camada de rede pode fornecer para seus usuários, é hora de ver como essa camada funciona por dentro. Duas organizações diferentes são possíveis, dependendo do tipo de serviço oferecido. Se o serviço sem conexão for oferecidos, os pacotes são injetados na rede individualmente e roteados independentemente um do outro. Nenhuma configuração prévia é necessária. Neste contexto, os pacotes são freqüentemente chamados de **datagramas** (em analogia com telegramas) e a rede é chamada **rede de datagramas**. Se o serviço orientado à conexão for usado, um caminho do roteador de origem até o roteador de destino deve ser estabelecido antes de qualquer pacotes de dados podem ser enviados. Esta conexão é chamada de **VC (circuito virtual)**, em analogia com os circuitos físicos configurados pelo sistema telefônico, e a rede é chamada **rede de circuito virtual**. Nesta seção, examinaremos a rede de datagramas trabalho; no próximo, examinaremos as redes de circuitos virtuais.

Vamos agora ver como funciona uma rede de datagramas. Suponha que o processo *P1* na Figura 5-2, há uma longa mensagem para *P2*. Ele passa a mensagem para a camada de transporte,

com instruções para entregá-lo ao processo *P2* no host *H2*. O código da camada de transporte é executado em *H1*, normalmente dentro do sistema operacional. Ele precede um cabeçalho de transporte

para a frente da mensagem e passa o resultado para a camada de rede, provavelmente apenas outro procedimento dentro do sistema operacional.

Mesa E
Mesa C's
Tabela de A (inicialmente) Tabela de A (mais tarde)
Dest. Linha
D
C
B
UMA
E
F
Pacote
Processo P1
Host H1

Roteador

Equipamento do ISP

H2

LAN

P2

4

2

3

1

UMA

B

B

-

C

C

D

B

E

C

F

C

UMA

B

B

-

C

C

D

B

E

B

F

B

UMA

B

UMA

UMA

C

-

D

E

E

F

E

UMA

B

D

C

C

D

D

E

-

F

F

Figura 5-2. Roteamento dentro de uma rede de datagramas.

Vamos supor para este exemplo que a mensagem seja quatro vezes mais longa do que tamanho máximo do pacote, então a camada de rede deve dividi-lo em quatro pacotes, 1, 2,

SEC. 5,1

QUESTÕES DE DESIGN DA CAMADA DE REDE

359

3 e 4, e envie cada um deles por sua vez para o roteador *A* usando algum pro ponto a ponto tocol, por exemplo, PPP. Nesse ponto, o ISP assume o controle. Cada roteador tem um tabela final informando para onde enviar os pacotes para cada um dos destinos possíveis. Cada entrada da tabela é um par que consiste em um destino e a linha de saída para usar para esse destino. Apenas linhas diretamente conectadas podem ser usadas. Por exemplo, em Figura 5-2, *A* tem apenas duas linhas de saída - para *B* e para *C* - portanto, todos os pacotes de entrada

deve ser enviado para um desses roteadores, mesmo que o destino final seja para algum outro roteador. A tabela de roteamento inicial de *A* é mostrada na figura sob o rótulo "ini-
cialmente."

Em *A*, os pacotes 1, 2 e 3 são armazenados brevemente, tendo chegado na entrada

link e tiveram suas somas de verificação verificadas. Em seguida, cada pacote é encaminhado de acordo

para *uma* tabela de, para a ligação de saída para *C* dentro de um novo quadro. O pacote 1 é então para-

desviou para *E* e, em seguida, a *F*. Quando chega a *F*, é enviado dentro de um quadro sobre o LAN para *H2*. Os pacotes 2 e 3 seguem a mesma rota.

No entanto, algo diferente acontece com o pacote 4. Quando chega a *A*, é enviado para o roteador *B*, embora também é destinado para *F*. Por algum motivo, *A* decidiu para enviar o pacote 4 por uma rota diferente daquela dos três primeiros pacotes. Talvez seja soube de um engarrafamento em algum lugar ao longo do caminho *ACE* e atualizou sua rotatabela de integração, conforme mostrado sob o rótulo "mais tarde." O algoritmo que gerencia as tabelas

e toma as decisões de roteamento é chamado de **algoritmo de roteamento**. Algoritmo de roteamento-

ritmos são um dos principais tópicos que estudaremos neste capítulo. Existem vários diferentes tipos deles, como veremos.

IP (Internet Protocol), que é a base de toda a Internet, é o exemplo importante de um serviço de rede sem conexão. Cada pacote carrega um destino Endereço IP que os roteadores usam para encaminhar individualmente cada pacote. Os endereços são 32 bits em pacotes IPv4 e 128 bits em pacotes IPv6. Descreveremos IP em muitos detalhes posteriormente neste capítulo.

5.1.4 Implementação de Serviço Orientado a Conexão

Para serviço orientado a conexão, precisamos de uma rede de circuito virtual. Deixe-nos ver como isso funciona. A ideia por trás dos circuitos virtuais é evitar ter que escolher uma nova rota para cada pacote enviado, como na Figura 5-2. Em vez disso, quando uma conexão é estabelecido, uma rota da máquina de origem para a máquina de destino é escolhida como parte da configuração da conexão e armazenados em tabelas dentro dos roteadores. Essa rota é usado para todo o tráfego que flui pela conexão, exatamente da mesma forma que o sistema telefônico funciona. Quando a conexão é liberada, o circuito virtual é também encerrado. Com o serviço orientado a conexão, cada pacote carrega uma identidade

fíx informando a qual circuito virtual ele pertence.

Como exemplo, considere a situação mostrada na Figura 5-3. Aqui, o host *H1* tem conexão estabelecida 1 com o host *H2*. Esta conexão é lembrada como a primeira entrada em cada uma das tabelas de roteamento. A primeira linha de *uma* tabela 's diz que se um pacote

Página 384

360

A CAMADA DE REDE
INDIVÍDUO. 5

identificador de conexão de rolamento 1 vem de *H1*, deve ser enviado ao roteador *C* e determinado identificador de conexão 1. Da mesma forma, a primeira entrada em *C* roteia o pacote para *E*,

também com identificador de conexão 1.

Mesa A
No
Fora
D
C
B
E
F
Pacote
Roteador
Equipamento do ISP
H2
LAN
P2
2
4
3
1
H1

```

H3 1
1
UMA
Processo P1
Host H1
P3
H3
C
C
2
1
Mesa C's
UMA
UMA
2
1
E
E
2
1
Mesa E
C
C
2
1
F
F
2
1

```

Figura 5-3. Roteamento em uma rede de circuito virtual.

Agora, vamos considerar o que acontece se *H3* também quiser estabelecer uma conexão para *H2*. Ele escolhe o identificador de conexão 1 (porque está iniciando a conexão e esta é sua única conexão) e informa a rede para estabelecer o circuito virtual. Isso leva à segunda linha nas tabelas. Observe que temos um conflito aqui por causa, embora *A* possa facilmente distinguir pacotes de conexão 1 de *H1* de conexões da conexão 1 de *H3*, *C* não podem fazer isso. Por este motivo, *A* atribui um diferente identificador de conexão para o tráfego de saída para a segunda conexão. Evitando conflitos desse tipo é porque os roteadores precisam da capacidade de substituir identificadores de conexão

fiers em pacotes de saída.

Em alguns contextos, esse processo é chamado de **troca de rótulo**. Um exemplo de O serviço de rede orientado para conexão é **MPLS** (**MultiProtocol Label Switching**). É usado em redes de ISP na Internet, com pacotes IP embrulhados em um Cabeçalho MPLS com um identificador ou rótulo de conexão de 20 bits. MPLS é frequentemente oculto de clientes, com o ISP estabelecendo conexões de longo prazo para grandes quantidades de tráfego, mas está cada vez mais sendo usado para ajudar quando a qualidade do serviço é importante, mas também com outras tarefas de gerenciamento de tráfego do ISP. Teremos mais a dizer sobre MPLS posteriormente neste capítulo.

5.1.5 Comparação de redes de circuitos virtuais e datagramas

Tanto os circuitos virtuais quanto os datagramas têm seus apoiadores e seus detratores. Agora tentaremos resumir os dois conjuntos de argumentos. Os principais problemas são listados na Fig. 5-4, embora os puristas provavelmente possam encontrar um contra-exemplo para tudo na figura.

Questão

Rede de datagrama

Rede de circuito virtual

Configuração de circuito

Não é necessário

Requeridos

Endereçando

Cada pacote contém o completo

endereço de origem e destino	
Cada pacote contém um	
número VC curto	
Informação do estado	
Roteadores não mantêm estado	
informações sobre conexões	
Cada VC requer roteador	
espaço de mesa por conexão	
Encaminhamento	
Cada pacote é roteado	
independente	
Rota escolhida quando VC é	
configuração; todos os pacotes o seguem	
Efeito das falhas do roteador	
Nenhum, exceto para pacotes	
perdido durante o acidente	
Todos os VCs que passaram	
através do falhado	
roteador está terminado	
Qualidade de serviço	
Difícil	
Fácil se houver recursos suficientes	
pode ser alocado em	
avanço para cada VC	
Controle de congestão	
Difícil	
Fácil se houver recursos suficientes	
pode ser alocado em	
avanço para cada VC	

Figura 5-4. Comparação de redes de datagramas e circuitos virtuais.

Dentro da rede, existem vários trade-offs entre circuitos virtuais e dados gramas. Uma compensação é o tempo de configuração versus o tempo de análise de endereço. Usando cir- virtual cuits requer uma fase de configuração, que leva tempo e consome recursos. Contudo, uma vez que este preço é pago, descobrir o que fazer com um pacote de dados em um circuito virtual rede cuit é fácil: o roteador apenas usa o número do circuito para indexar em uma tabela para descubra para onde vai o pacote. Em uma rede de datagramas, nenhuma configuração é necessária, mas um um procedimento de pesquisa mais complicado é necessário para localizar a entrada para o destino ção.

Um problema relacionado é que os endereços de destino usados nas redes de datagramas são mais do que os números de circuito usados em redes de circuito virtual porque eles têm um significado global. Se os pacotes tendem a ser bastante curtos, incluindo um destino completo endereço em cada pacote pode representar uma quantidade significativa de sobrecarga, e portanto, um desperdício de largura de banda.

Outro problema é a quantidade de espaço de mesa necessária na memória do roteador. UMA rede de datagramas precisa ter uma entrada para cada destino possível, enquanto um A rede de circuito virtual precisa apenas de uma entrada para cada circuito virtual. No entanto, este

vantagem é um tanto ilusória, uma vez que os pacotes de configuração de conexão devem ser roteados também, e eles usam endereços de destino, da mesma forma que os datagramas fazem. Os circuitos virtuais apresentam algumas vantagens na garantia da qualidade do serviço e evitando congestionamento na rede porque os recursos (por exemplo, buffers, banda largura e ciclos de CPU) podem ser reservados com antecedência, quando a conexão é estabelecida. Assim que os pacotes começarem a chegar, a largura de banda necessária e a capacidade do roteador já estarão lá. Com uma rede de datagramas, evitar congestionamento é mais difícil custo.

Para sistemas de processamento de transações (por exemplo, lojas ligando para verificar o cartão de crédito compras), a sobrecarga necessária para configurar e limpar um circuito virtual pode facilmente anão o uso do circuito. Se a maior parte do tráfego for deste

tipo, o uso de circuitos virtuais dentro da rede faz pouco sentido. No outro lado, para usos de longa duração, como tráfego VPN entre dois escritórios corporativos, circuitos virtuais permanentes (que são configurados manualmente e duram meses ou anos) pode ser útil.

Os circuitos virtuais também apresentam um problema de vulnerabilidade. Se um roteador travar e perde a memória, mesmo que volte um segundo depois, todos os circuitos virtuais passando por ele terá que ser abortado. Em contraste, se um roteador de datagrama for desativado, apenas os usuários cujos pacotes estavam enfileirados no roteador no momento precisam sofrer (e provavelmente nem mesmo assim, uma vez que o remetente provavelmente os retransmitirá Em breve). A perda de uma linha de comunicação é fatal para os circuitos virtuais que a utilizam, mas

pode ser facilmente compensado se datagramas forem usados. Os datagramas também permitem o roteadores para equilibrar o tráfego em toda a rede, uma vez que as rotas podem ser alteradas no meio de uma longa sequência de transmissões de pacotes.

5.2 ALGORITMOS DE ROTEAMENTO

A principal função da camada de rede é o roteamento de pacotes da fonte machine para a máquina de destino. Na maioria das redes, os pacotes exigirão vários saltos para fazer a viagem. A única exceção notável é para redes de transmissão, mas mesmo aqui o roteamento é um problema se a origem e o destino não estiverem no mesmo segmento de rede. Os algoritmos que escolhem as rotas e as estruturas de dados que eles usam são uma área importante do projeto da camada de rede.

O **algoritmo de roteamento** é a parte do software da camada de rede responsável para decidir em qual linha de saída um pacote de entrada deve ser transmitido. E se a rede usa datagramas internamente, esta decisão deve ser tomada novamente a cada pacote de dados chegando desde a melhor rota pode ter mudado desde a última vez. Se o rede usa circuitos virtuais internamente, as decisões de roteamento são feitas apenas quando um novo circuito virtual está sendo configurado. Depois disso, os pacotes de dados apenas seguem o já rota estabelecida. O último caso às vezes é chamado de **roteamento de sessão** porque um a rota permanece em vigor por uma sessão inteira (por exemplo, enquanto estiver conectado em uma VPN).

Página 387

SEC. 5.2

ALGORITMOS DE ROTEAMENTO

363

Às vezes é útil fazer uma distinção entre o roteamento, o que significa a decisão de quais rotas usar e o encaminhamento, que é o que acontece quando um o pacote chega. Pode-se pensar em um roteador como tendo dois processos dentro dele. 1 deles trata cada pacote à medida que chega, procurando a linha de saída para usar para ele nas tabelas de roteamento. Este processo está **encaminhando**. O outro processo é responsável capaz de preencher e atualizar as tabelas de roteamento. É aí que o algoritmo de roteamento ritmo entra em jogo.

Independentemente de as rotas serem escolhidas independentemente para cada pacote enviado ou somente quando novas conexões são estabelecidas, certas propriedades são desejáveis em um algoritmo de roteamento: correção, simplicidade, robustez, estabilidade, justiça e eficácia ciência. Correção e simplicidade dificilmente exigem comentários, mas a necessidade de a robustez pode ser menos óbvia no início. Assim que uma grande rede entra no ar, pode ser esperado para funcionar continuamente por anos sem falhas em todo o sistema. Durante esse período, haverá falhas de hardware e software de todos os tipos. Hosts, roteadores e linhas falharão repetidamente e a topologia mudará muitas vezes.

O algoritmo de roteamento deve ser capaz de lidar com mudanças na topologia e tráfego sem exigir que todos os trabalhos em todos os hosts sejam cancelados. Imagine a confusão se a rede precisava ser reiniciada toda vez que algum roteador travava!

A estabilidade também é uma meta importante para o algoritmo de roteamento. Existe rotas-algoritmos que nunca convergem para um conjunto fixo de caminhos, não importa quanto tempo eles

corre. Um algoritmo estável atinge o equilíbrio e permanece lá. Deve convergir rapidamente também, uma vez que a comunicação pode ser interrompida até o algoritmo de roteamento atingiu o equilíbrio.

Justiça e eficiência podem parecer óbvias - certamente nenhuma pessoa razoável se oporia a eles - mas, ao que parece, eles costumam ser objetivos contraditórios. Como um exemplo simples desse conflito, veja a Figura 5-5. Suponha que haja o suficiente tráfego entre A e A' , entre B e B' , e entre C e C' para saturar o links horizontais. Para maximizar o fluxo total, o tráfego de X para X' deve ser desligado completamente. Infelizmente, X e X' podem não ver dessa forma. Evidentemente, alguns compromisso entre eficiência global e justiça para conexões individuais é necessário.

Antes mesmo de tentarmos encontrar soluções de compromisso entre justiça e eficiência, devemos decidir o que procuramos otimizar. Minimizando o atraso médio do pacote é um candidato óbvio para enviar tráfego pela rede de maneira eficaz, mas também é maximizando o rendimento total da rede. Além disso, esses dois objetivos também estão em conflito, uma vez que operar qualquer sistema de filas próximo à capacidade implica em uma longa fila

atraso ing. Como um meio-termo, muitas redes tentam minimizar a distância de um o pacote deve viajar ou simplesmente reduzir o número de saltos que um pacote deve fazer. Esta escolha tende a melhorar o atraso e também reduzir a quantidade de largura de banda consumido por pacote, o que tende a melhorar o rendimento geral da rede conforme bem.

Os algoritmos de roteamento podem ser agrupados em duas classes principais: não adaptativos e adaptativo. **Algoritmos não adaptativos** não baseiam suas decisões de roteamento em qualquer

364

A CAMADA DE REDE INDIVÍDUO. 5

X
X'
UMA
B
C
UMA'
B'
C'

Figura 5-5. Rede com conflito entre justiça e eficiência.

medidas ou estimativas da topologia e tráfego atuais. Em vez disso, a escolha da rota a ser usada para ir de I a J (para todos I e J) é calculado antecipadamente, off-line e baixado para os roteadores quando a rede é inicializada. Este procedimento às vezes é chamado de **roteamento estático**. Porque não responde a falhas, estático o roteamento é útil principalmente para situações em que a escolha do roteamento é clara. Para exemplo, o roteador F na Figura 5-3 deve enviar pacotes dirigidos à rede para o roteador E independentemente do destino final.

Os algoritmos adaptativos, por outro lado, mudam suas decisões de roteamento para refletir mudanças na topologia e, às vezes, mudanças no tráfego também. Estes algoritmos de **roteamento dinâmico** diferem em onde eles obtêm suas informações (por exemplo, localmente, de roteadores adjacentes ou de todos os roteadores), quando eles mudam as rotas (por exemplo, quando a topologia muda, ou a cada ΔT segundos conforme a carga muda), e qual métrica é usada para otimização (por exemplo, distância, número de saltos ou estimativa tempo de trânsito).

Nas seções a seguir, discutiremos uma variedade de algoritmos de roteamento. Os algoritmos cobrem modelos de entrega, além de enviar um pacote de uma fonte para um destino. Às vezes, o objetivo é enviar o pacote para vários, todos ou um de um conjunto de destinos. Todos os algoritmos de roteamento que descrevemos aqui tomam decisões

com base na topologia; nós adiamos a possibilidade de decisões com base no tráfego níveis da Seção 5.3.

5.2.1 O Princípio da Otimidade

Antes de entrarmos em algoritmos específicos, pode ser útil observar que se pode fazer uma declaração geral sobre as rotas ideais sem levar em conta o topo da rede logy ou tráfego. Esta afirmação é conhecida como o **princípio da otimização** (Bellman, 1957). Ele afirma que se o roteador J estiver no caminho ideal do roteador I para o roteador K ,

SEC. 5,2

ALGORITMOS DE ROTEAMENTO

365

então, o caminho ótimo de J para K também segue a mesma rota. Para ver isso, ligue a parte da rota de I a J r_1 e o resto da rota r_2 . Se uma rota melhor do que r_2 existia de J a K , ele poderia ser concatenado com r_1 para melhorar a rota de I a K , contradizendo nossa afirmação de que $r_1 r_2$ é ótimo.

Como consequência direta do princípio da otimalidade, podemos ver que o conjunto de as rotas ideais de todas as fontes para um determinado destino formam uma árvore enraizada no destino. Essa árvore é chamada de **árvore sumidouro** e é ilustrada na Fig. 5-6 (b), onde a métrica de distância é o número de saltos. O objetivo de todo algoritmo de roteamento ritmos é descobrir e usar as árvores dissipadoras para todos os roteadores.

B

UMA

F

D

E

C

J

N

O

Eu

H

G

eu

M

K

(uma)

B

UMA

F

D

E

C

J

N

O

Eu

H

G

eu

M

K

(b)

Figura 5-6. (a) Uma rede. (b) Uma árvore de pia para o roteador B .

Observe que uma árvore sink não é necessariamente única; outras árvores com o mesmo caminho podem existir comprimentos. Se permitirmos que todos os caminhos possíveis sejam escolhidos, a árvore será

vem uma estrutura mais geral chamada **DAG** (**Directed Acyclic Graph**). DAGs não tem loops. Usaremos árvores afundar como uma abreviatura conveniente para ambos os casos. Ambos os casos também dependem da suposição técnica de que os caminhos não interferem uns com os outros, por exemplo, um engarrafamento em um caminho não causará outro caminho para desviar.

Uma vez que uma árvore sink é de fato uma árvore, ela não contém nenhum loop, então cada pacote será entregue dentro de um número finito e limitado de saltos. Na prática, a vida é não é tão fácil. Links e roteadores podem cair e voltar a funcionar durante a operação ção, então roteadores diferentes podem ter idéias diferentes sobre a topologia atual.

Além disso, resolvemos discretamente a questão de saber se cada roteador deve individualmente adquirir as informações sobre as quais basear seu cálculo da árvore sink ou se este

as informações são coletadas por outros meios. Voltaremos a essas questões Em breve. No entanto, o princípio de optimalidade e a árvore de dissipação fornecem uma marca contra a qual outros algoritmos de roteamento podem ser medidos.

366

A CAMADA DE REDE
INDIVÍDUO. 5

5.2.2 Algoritmo do Caminho Mais Curto

Vamos começar nosso estudo de algoritmos de roteamento com uma técnica simples para Colocando caminhos ótimos dado um quadro completo da rede. Esses caminhos são os aqueles que queremos que um algoritmo de roteamento distribuído encontre, embora nem todos os ers podem saber todos os detalhes da rede.

A ideia é construir um grafo da rede, com cada nó do grafo representando um roteador e cada borda do gráfico representando uma comunicação linha ou link. Para escolher uma rota entre um determinado par de roteadores, o algoritmo apenas encontra o caminho mais curto entre eles no gráfico.

O conceito de **caminho mais curto** merece alguma explicação. Uma forma de o comprimento do caminho de medição é o número de saltos. Usando essa métrica, os caminhos *ABC*

e *ABE* na Fig. 5-7 são igualmente longos. Outra métrica é a distância geográfica em quilômetros, caso em que *ABC* é claramente muito mais longo do que *ABE* (assumindo o figura é desenhada em escala).

UMA

D

1

2

6

G

4

(uma)

F (∞ , -)

D (∞ , -)

UMA

B

7

C

2

H

3

3

2

2

F

E

1

2

2

6

G

4

UMA

(c)

UMA

B (2, A)

C (9, B)

H (∞ , -)

E (4, B)

G (6, A)

F (6, E)

D (∞ , -)

UMA

(e)

UMA

B (2, A)

C (9, B)

H (9, G)

E (4, B)

G (5, E)

F (6, E)

D (∞ , -)

UMA

(f)

```

UMA
B (2, A)
C (9, B)
H (8, F)
E (4, B)
G (5, E)
F (6, E)
D (∞, 1)
UMA
(d)
UMA
B (2, A)
C (9, B)
H (∞, -)
E (4, B)
G (5, E)
F (∞, -)
D (∞, -)
UMA
H
E
G
(b)
B (2, A)
C (∞, -)
H (∞, -)
E (∞, -)
G (6, A)

```

Figura 5-7. Os primeiros seis passos utilizados no cálculo do caminho mais curto de *um* a *D*.
As setas indicam o nó de trabalho.

Página 391

SEC. 5,2

ALGORITMOS DE ROTEAMENTO

367

No entanto, muitas outras métricas além de saltos e distância física também são possível. Por exemplo, cada borda pode ser rotulada com o atraso médio de um padrão pacote de teste, conforme medido por execuções de hora em hora. Com este rótulo de gráfico, o mais curto

path é o caminho mais rápido, em vez do caminho com menos bordas ou quilômetros.

No caso geral, os rótulos nas bordas podem ser calculados como uma função de a distância, largura de banda, tráfego médio, custo de comunicação, atraso medido, e outros fatores. Ao alterar a função de ponderação, o algoritmo então computar o caminho "mais curto" medido de acordo com qualquer um de uma série de critérios ou a uma combinação de critérios.

Vários algoritmos para calcular o caminho mais curto entre dois nós de um gráfico são conhecidos. Este é devido a Dijkstra (1959) e encontra os caminhos mais curtos entre uma origem e todos os destinos da rede. Cada nó é rotulado (em parênteses) com sua distância do nó de origem ao longo do caminho mais conhecido.

As distâncias devem ser não negativas, como serão se forem baseadas em quan- como largura de banda e atraso. Inicialmente, nenhum caminho é conhecido, então todos os nós são rotulado com infinito. À medida que o algoritmo prossegue e os caminhos são encontrados, os rótulos

pode mudar, refletindo melhores caminhos. Um rótulo pode ser provisório ou permanente.

Inicialmente, todos os rótulos são provisórios. Quando é descoberto que um rótulo representa o caminho mais curto possível da fonte para esse nó, ele se torna permanente e nunca mudou depois disso.

Para ilustrar como o algoritmo de rotulagem funciona, olhe para o peso, gráfico não direcionado da Fig. 5-7 (a), onde os pesos representam, por exemplo, distância. Queremos encontrar o caminho mais curto de *A* a *D*. Começamos marcando nó *A* como permanente, indicado por um círculo preenchido. Em seguida, examinamos, por sua vez, cada um dos nós adjacentes a *A* (o nó de trabalho), renomeando cada um com o distanciar para *A*. Sempre que um nó é renomeado, também o rotulamos com o nó de qual a sonda foi feita para que possamos reconstruir o caminho final mais tarde. Se a rede tinha mais de um caminho mais curto de *A* a *D* e queríamos encontrar todos

eles, precisaríamos nos lembrar de todos os nós de sondagem que poderiam alcançar um nó com a mesma distância.

Tendo examinado cada um dos nós adjacentes a A , examinamos todos os tenta-nós rotulados de forma ativa em todo o gráfico e fazer aquele com o menor rótulo permanente, conforme mostrado na Figura 5-7 (b). Este se torna o novo nó de trabalho.

Agora começamos em B e examinamos todos os nós adjacentes a ele. Se a soma do rótulo em B e a distância de B ao nó sendo considerado é menor do que o rótulo em esse nó, temos um caminho mais curto, então o nó é renomeado.

Depois que todos os nós adjacentes ao nó de trabalho foram inspecionados e o rótulos provisórios alterados se possível, todo o gráfico é pesquisado para o nó rotulado com o menor valor. Este nó se torna permanente e se torna o nó de trabalho para a próxima rodada. A Figura 5-7 mostra as primeiras seis etapas do algoritmo.

Para ver por que o algoritmo funciona, observe a Figura 5-7 (c). Neste ponto nós temos acabou de tornar E permanente. Suponha que haja um caminho mais curto do que ABE , digamos

Página 392

368

A CAMADA DE REDE INDIVÍDUO. 5

$AXYZE$ (para alguns X e Y). Existem duas possibilidades: qualquer um dos nós Z já foi tornado permanente, ou não foi. Se tiver, então E já foi sondado (na rodada seguinte àquela em que Z se tornou permanente), então o caminho $AXYZE$ não escapou à nossa atenção e, portanto, não pode ser um caminho mais curto.

Agora considere o caso em que Z ainda está provisoriamente rotulado. Se o rótulo em Z for maior ou igual àquele em E , então $AXYZE$ não pode ser um caminho mais curto que ABE . Se o rótulo for inferior ao de E , então Z e não E se tornará permanente primeiro, allowing E para ser sondado de Z .

Esse algoritmo é apresentado na Figura 5.8. As variáveis globais n e $dist$ descrevem o gráfico e são inicializados antes que o *caminho mais curto* seja chamado. A única diferença entre o programa e o algoritmo descrito acima é que na Fig. 5-8, nós calcule o caminho mais curto começando no nó terminal, t , ao invés de na fonte nó, s .

Uma vez que os caminhos mais curtos de t para s em um gráfico não direcionado são os mesmos que os caminhos mais curtos de s para t , não importa em que extremidade começamos. O motivo para pesquisar para trás é que cada nó é rotulado com seu predecessor, em vez do seu sucessor. Quando o caminho final é copiado para a variável de saída, *caminho*, o caminho é, portanto, invertido. Os dois efeitos de reversão se cancelam e a resposta é produzido na ordem correta.

5.2.3 Inundações

Quando um algoritmo de roteamento é implementado, cada roteador deve tomar decisões com base no conhecimento local, não no quadro completo da rede. Um simples técnica local está **inundando**, em que cada pacote de entrada é enviado em cada linha de saída, exceto aquela em que chegou.

A inundação obviamente gera um grande número de pacotes duplicados, na verdade, um número infinito, a menos que algumas medidas sejam tomadas para amortecer o processo. Um tal medida é ter um contador de saltos contido no cabeçalho de cada pacote que é diminuiu a cada salto, com o pacote sendo descartado quando o contador chega a zero. Idealmente, o contador de saltos deve ser inicializado com o comprimento do caminho da origem ao destino. Se o remetente não souber a extensão do caminho, ele pode inicializar o contador para o pior caso, ou seja, o diâmetro total da rede trabalhos.

Inundar com uma contagem de saltos pode produzir um número exponencial de duplicatas pacotes conforme a contagem de saltos aumenta e os roteadores duplicam os pacotes que viram diante. Uma técnica melhor para represar a inundação é fazer com que os roteadores acompanhem quais pacotes foram inundados, para evitar enviá-los uma segunda vez. 1

maneira de atingir este objetivo é fazer com que o roteador de origem coloque um número de sequência em cada pacote que recebe de seus hosts. Cada roteador precisa de uma lista por fonte roteador informando quais números de sequência originados nessa fonte já foi visto. Se um pacote recebido estiver na lista, ele não será inundado.

Página 393

SEC. 5,2
ALGORITMOS DE ROTEAMENTO

369

```
# define MAX NODES 1024
/* número máximo de nós */
#define INFINITY 1000000000
/* um número maior do que cada caminho máximo */
int n, dist [MAX NODES] [MAX NODES];
/* dist [i] [j] é a distância de i a j */
anular o caminho mais curto (int s, int t, int path [])
{struct state {
    /* o caminho que está sendo trabalhado */
    predecessor int;
    /* nó anterior */
    comprimento interno;
    /* comprimento da fonte até este nó */
    etiqueta enum {permanente, tentativa};
    /* label state */
    } estado [MAX NODES];
int i, k, min;
estado da estrutura * p;
para (p = & estado [0]; p <& estado [n]; p++) {
    /* estado de inicialização */
    p-> predecessor = -1;
    p-> comprimento = INFINITO;
    p-> rótulo = tentativa;
}
estado [t]. comprimento = 0; estado [t]. etiqueta = permanente;
k = t;
/* k é o nó de trabalho inicial */
Faz {
    /* Existe um caminho melhor de k? */
    para (i = 0; i <n; i++)
        /* este gráfico possui n nós */
        if (dist [k] [i]! = 0 && estado [i]. label == tentativa) {
            if (estado [k]. comprimento + dist [k] [i] <estado [i]. comprimento) {
                estado [i]. predecessor = k;
                estado [i]. comprimento = estado [k]. comprimento + dist [k] [i];
            }
        }
    /* Encontre o nó provisoriamente rotulado com o menor rótulo. */
    k = 0; min = INFINITO;
    para (i = 0; i <n; i++)
        if (estado [i]. label == tentativa && estado [i]. comprimento <min) {
            min = estado [i]. comprimento;
            k = i;
        }
    estado [k]. etiqueta = permanente;
    } enquanto (k! = s);
    /* Copie o caminho para a matriz de saída. */
    i = 0; k = s;
    faça {caminho [i ++] = k; k = estado [k] .predecessor;} enquanto (k> = 0);
}
```

Figura 5-8. O algoritmo de Dijkstra para calcular o caminho mais curto em um gráfico.

Para evitar que a lista cresça sem limites, cada lista deve ser agostado

mentado por um contador, k , o que significa que todos os números de sequência até k foram

visto. Quando um pacote chega, é fácil verificar se o pacote já foi

370

A CAMADA DE REDE

INDIVÍDUO. 5

inundado (comparando seu número de sequência com k ; em caso afirmativo, ele é descartado. mais, a lista completa abaixo de k não é necessária, uma vez que k efetivamente a resume. A inundação não é prática para enviar a maioria dos pacotes, mas tem algum impacto importantes usos. Primeiro, ele garante que um pacote seja entregue a todos os nós da rede trabalhos. Isso pode ser um desperdício se houver um único destino que precisa do pacote, mas é eficaz para transmitir informações. Em redes sem fio, todas as mensagens sábios transmitidos por uma estação podem ser recebidos por todas as outras estações de seu rádio intervalo, que é, na verdade, inundação, e alguns algoritmos utilizam essa propriedade. Em segundo lugar, as inundações são tremendamente robustas. Mesmo que um grande número de roteadores sejam explodido em bits (por exemplo, em uma rede militar localizada em uma zona de guerra), a inundação encontrará um caminho, se houver, para levar um pacote ao seu destino. As inundações também exigem pouco na forma de configuração. Os roteadores precisam apenas conhecer seus vizinhos. Isso significa que a inundação pode ser usada como um bloco de construção para outros algoritmos de roteamento que são mais eficiente, mas precisa de mais na forma de configuração. A inundação também pode ser usada como um métrica com a qual outros algoritmos de roteamento podem ser comparados. Inundando sempre escolhe o caminho mais curto porque escolhe todos os caminhos possíveis em paralelo. Vigarista-sequencialmente, nenhum outro algoritmo pode produzir um atraso menor (se ignorarmos o gerada pelo próprio processo de inundação).

5.2.4 Roteamento de vetor de distância

Redes de computadores geralmente usam algoritmos de roteamento dinâmico que são mais complexos do que inundações, mas mais eficientes porque encontram os caminhos mais curtos para o

topologia atual. Dois algoritmos dinâmicos em particular, roteamento de vetor de distância e roteamento de estado de link, são os mais populares. Nesta seção, veremos o algoritmo anterior. Na seção seguinte, estudaremos o último algoritmo.

Um algoritmo de **roteamento do vetor de distância** opera fazendo com que cada roteador mantenha uma tabela (ou seja, um vetor) dando a melhor distância conhecida para cada destino e qual link usar para chegar lá. Estas tabelas são atualizadas pela troca de informações com os vizinhos. Eventualmente, cada roteador conhece o melhor link para alcançar cada destino.

O algoritmo de roteamento do vetor de distância às vezes é chamado por outros nomes, mais comumente, o **algoritmo de roteamento Bellman-Ford** distribuído, após o pesquisadores que o desenvolveram (Bellman, 1957; e Ford e Fulkerson, 1962). isto era o algoritmo de roteamento ARPANET original e também era usado na Internet sob o nome de RIP.

No roteamento do vetor de distância, cada roteador mantém uma tabela de roteamento indexada por, e contendo uma entrada para cada roteador na rede. Esta entrada tem duas partes: a linha de saída preferida para usar para esse destino e uma estimativa do dis- para esse destino. A distância pode ser medida como o número de saltos ou usando outra métrica, como discutimos para calcular os caminhos mais curtos. Presume-se que o roteador conheça a "distância" de cada um de seus vizinhos. Se o métrica é o salto, a distância é de apenas um salto. Se a métrica for o atraso de propagação, o

SEC. 5,2

ALGORITMOS DE ROTEAMENTO

roteador pode medi-lo diretamente com pacotes ECHO especiais que o receptor apenas marca o tempo e envia de volta o mais rápido possível.

Por exemplo, suponha que o atraso seja usado como uma métrica e que o roteador conhece a demora para cada um de seus vizinhos. Uma vez a cada T ms, cada roteador envia para cada vizinho uma lista de seus atrasos estimados para cada destino. Ele também recebe um lista semelhante de cada vizinho. Imagine que uma dessas mesas acabou de entrar do vizinho X , com X_i sendo a estimativa de X de quanto tempo leva para chegar ao roteador i . Se o roteador sabe que o atraso de X é m ms, ele também sabe que pode alcançar roteador i via X em $X_i + m$ mseg. Ao realizar este cálculo para cada vizinho, um roteador pode descobrir qual estimativa parece ser a melhor e usar essa estimativa e o link correspondente em sua nova tabela de roteamento. Observe que a velha tabela de roteamento não é usada no cálculo.

Esse processo de atualização é ilustrado na Figura 5-9. A parte (a) mostra uma rede. As primeiras quatro colunas da parte (b) mostram os vetores de atraso recebidos dos vizinhos de roteador J . A afirma ter um atraso de 12 ms para B , um atraso de 25 ms para C , um 40-atraso de ms para D , etc. Suponha que J tenha medido ou estimado seu atraso para seu vizinhos, A , I , H e K , como 8, 10, 12 e 6 mseg, respectivamente.

```
(uma)
UMA
B
C
D
E
Eu
J
K
eu
F
G
H
Roteador
0
12
25
40
14
23
18
17
21
9
24
29
24
36
18
27
7
20
31
20
0
11
22
33
20
31
19
8
30
19
6
0
14
7
22
9
21
28
36
24
22
40
31
19
```

```

22
10
0
9
8
20
28
20
17
30
18
12
10
0
6
15
UMA
UMA
Eu
H
Eu
Eu
H
H
Eu
-
K
K
Para A
Eu
H
K
Linha
Nova estimativa
atraso de J
UMA
B
C
D
E
F
G
H
Eu
J
K
eu
JA
JI
JH
JK
demora
demora
demora
demora
é
é
é
é
8
10
12
6
Novo
roteamento
tabela
para J
Vetores recebidos de
Quatro vizinhos de J
(b)

```

Figura 5-9. (a) Uma rede. (b) de entrada a partir de um , I , H , K , e a nova tabela de encaminhamento para J .

Considere como J calcula a sua nova rota para roteador G . Sabe que pode chegar a

A em 8 ms e, além disso, A afirma ser capaz de chegar a G em 18 ms, então J

sabe que pode contar com um atraso de 26 ms para G se encaminhar pacotes com destino a G

(6 + 12) e 37 (31 + 6) ms, respectivamente. O melhor desses valores é 18, então faz uma entrada em sua tabela de roteamento que o atraso para G é de 18 ms e que a rota para utilização é através H . O mesmo cálculo é executado para todos os outros destinos, com a nova tabela de roteamento mostrada na última coluna da figura.

O problema da contagem ao infinito

O estabelecimento de rotas para os melhores caminhos na rede é chamado de **convergência**. O roteamento do vetor de distância é útil como uma técnica simples pela qual os roteadores podem calcular

calcula os caminhos mais curtos de forma seletiva, mas tem uma séria desvantagem na prática: al-embora convirja para a resposta correta, pode fazê-lo lentamente. Em particular, reage rapidamente às boas notícias, mas vagarosamente às más notícias. Considere um roteador cujo a melhor rota para o destino X é longa. Se, na próxima troca, o vizinho A de repente relata um pequeno atraso para X , o roteador apenas muda para usar a linha para A para enviar tráfego para X . Em uma troca de vetor, as boas notícias são processadas.

Para ver com que rapidez as boas notícias se propagam, considere a rede de cinco nós (linear) trabalho da Fig. 5-10, onde a métrica de atraso é o número de saltos. Suponha que A seja desativado inicialmente e todos os outros roteadores sabem disso. Em outras palavras, eles têm todos registrado o atraso para A como infinito.

UMA
B
C
D
E
•
•
•
•
•
•
•
•
4
1
1
1
1
2
2
2
3
3
Inicialmente
Após 1 troca
Após 2 trocas
Após 3 trocas
Após 4 trocas

UMA
B
C
D
E
1
2
3
4
•
•
•
•
2
3
4
3
4
4
6
3
3
5
5
4

4
6
5
5
6
7
6
7
7
8
8
7
Inicialmente
Após 1 troca
Após 2 trocas
Após 3 trocas
Após 4 trocas
Após 5 trocas
Após 6 trocas
...
(uma)
(b)

Figura 5-10. O problema da contagem até o infinito.

Quando A surge, os outros roteadores aprendem sobre ele por meio das trocas de vetores. Para simplificar, vamos supor que existe um gong gigante em algum lugar que está golpeado periodicamente para iniciar uma troca de vetor em todos os roteadores simultaneamente. Em o momento da primeira troca, B aprende que seu vizinho esquerdo tem atraso zero para um . B agora faz uma entrada em sua tabela de roteamento indicando que A está a um salto de distância para a esquerda. Todos os outros roteadores ainda pensam que A está inativo. Neste ponto, a rota As entradas da tabela de referência para A são mostradas na segunda linha da Figura 5-10 (a). Na proxima

Página 397

SEC. 5.2

ALGORITMOS DE ROTEAMENTO

373

troca, C aprende que B tem um caminho de comprimento 1 para A , então ele atualiza sua tabela de roteamento

para indicar um caminho de comprimento 2, mas D e E não ouvem as boas novas até mais tarde. Claramente, as boas novas estão se espalhando à taxa de um salto por troca. Em uma rede trabalho cujo caminho mais longo é de comprimento N saltos, dentro de N trocas todos irão saber sobre links e roteadores recentemente revividos.

Agora, vamos considerar a situação da Fig. 5-10 (b), em que todos os links e os roteadores estão inicialmente ativos. Os roteadores B , C , D e E têm distâncias para A de 1, 2, 3 e 4 saltos, respectivamente. De repente, ou A cai ou o link entre A e B é corte (que é efetivamente a mesma coisa do ponto de vista de B).

Na primeira troca de pacotes, B não ouvir nada do A . Felizmente, C diz " Não se preocupe; Eu tenho um caminho para A de comprimento 2. " B pouco suspeita que C 's o caminho passa pelo próprio B . Pelo que B sabe, C pode ter dez links, todos com classifique os caminhos para A de comprimento 2. Como resultado, B pensa que pode alcançar A via C , com um caminho

comprimento de 3. D e E não atualizam suas entradas para A na primeira troca.

Na segunda troca, C percebe que cada um de seus vizinhos afirma ter um caminho para A de comprimento 3. Ele escolhe um deles ao acaso e faz sua nova distância a A , conforme mostrado na terceira linha da Fig. 5-10 (b). As trocas subsequentes produzem a história mostrada no restante da Figura 5.10 (b).

A partir desta figura, deve ficar claro por que as más notícias viajam lentamente: nenhum roteador sempre tem um valor maior do que o mínimo de todos os seus vizinhos.

Gradualmente, todos os roteadores seguem seu caminho até o infinito, mas o número de trocas necessário depende do valor numérico usado para infinito. Por esse motivo, é sábio definir infinito para o caminho mais longo mais 1.

Não é totalmente surpreendente que este problema seja conhecido como probabilidade de **contagem até o infinito**

lem. Houve muitas tentativas de resolvê-lo, por exemplo, evitando roteadores de anunciar seus melhores caminhos de volta aos vizinhos de quem ouviram eles com o horizonte dividido com regra reversa envenenada discutida na RFC 1058. No entanto, nenhuma dessas heurísticas funciona bem na prática, apesar do colorido nomes. O cerne do problema é que quando X diz a Y que ele tem um caminho onde, Y não tem como saber se ele próprio está no caminho.

5.2.5 Roteamento de Estado de Link

O roteamento do vetor de distância foi usado na ARPANET até 1979, quando foi substituído pelo roteamento de estado de link. O principal problema que causou sua morte foi que o algoritmo costuma demorar muito para convergir após a topologia da rede alterado (devido ao problema da contagem até o infinito). Consequentemente, foi substituído por um algoritmo inteiramente novo, agora chamado de **roteamento de estado de link**. Variantes de link state

roteamento chamado IS-IS e OSPF são os algoritmos de roteamento que são mais amplamente usado em grandes redes e na Internet hoje.

A ideia por trás do roteamento de estado de link é bastante simples e pode ser definida como cinco partes. Cada roteador deve fazer o seguinte para funcionar:

Página 398

374

A CAMADA DE REDE

INDIVÍDUO. 5

1. Descubra seus vizinhos e aprenda seus endereços de rede.
2. Defina a distância ou métrica de custo para cada um de seus vizinhos.
3. Construa um pacote contando tudo o que acabou de aprender.
4. Envie este pacote e receba pacotes de todos os outros roteadores.
5. Calcule o caminho mais curto para todos os outros roteadores.

Na verdade, a topologia completa é distribuída para cada roteador. Então o algoritmo de Dijkstra pode ser executado em cada roteador para encontrar o caminho mais curto para todos os outros roteadores.

A seguir, consideraremos cada uma dessas cinco etapas com mais detalhes.

Aprendendo sobre os vizinhos

Quando um roteador é inicializado, sua primeira tarefa é descobrir quem são seus vizinhos. Isto atinge esse objetivo enviando um pacote especial HELLO em cada ponto a ponto.

Espera-se que o roteador na outra extremidade envie uma resposta com seu nome.

Esses nomes devem ser globalmente únicos, porque quando um roteador distante ouve mais tarde que três roteadores estão todos conectados a F , é essencial que ele possa determinar se er todos os três o mesmo significado F .

Quando dois ou mais roteadores estão conectados por um link de transmissão (por exemplo, um switch, anel ou Ethernet clássica), a situação é um pouco mais complicada. Fig. 5-11 (a) ilustra uma LAN de transmissão para a qual três roteadores, A , C e F , estão diretamente conectados. Cada um desses roteadores está conectado a um ou mais roteadores adicionais, como mostrando.

Roteador
UMA
B
C
D
E
C
D
E
H
Eu
F
G
G
H
Eu

F
N
UMA
B
LAN
(uma)
(b)

Figura 5-11. (a) Nove roteadores e uma LAN de transmissão. (b) Um modelo gráfico de (a). A transmissão LAN fornece conectividade entre cada par de roteamento conectado. No entanto, modelar a LAN com tantos links ponto a ponto aumenta o tamanho

Página 399

SEC. 5,2
ALGORITMOS DE ROTEAMENTO

375

da topologia e leva a mensagens inúteis. Uma maneira melhor de modelar a LAN é considerá-lo um nó em si, como mostra a Figura 5.11 (b). Aqui, temos a introdução produziu um novo nó artificial, N , ao qual A , C e F estão conectados. Um **design**-

O roteador conectado na LAN é selecionado para desempenhar a função de N no protocolo de roteamento.

O fato de ser possível ir de A para C na LAN é representado pelo caminho ANC aqui.

Definindo custos de link

O algoritmo de roteamento de estado de link requer que cada link tenha uma distância ou custo métrica para encontrar os caminhos mais curtos. O custo para alcançar os vizinhos pode ser definido automaticamente, ou configurado pela operadora de rede. Uma escolha comum é fazer o custo inversamente proporcional à largura de banda do link. Por exemplo, 1 Gbps Ethernet pode ter um custo de 1 e Ethernet de 100 Mbps um custo de 10. Isso torna caminhos de maior capacidade melhores escolhas.

Se a rede estiver distribuída geograficamente, o atraso dos links pode ser factored no custo de modo que os caminhos em links mais curtos sejam escolhas melhores. A maioria maneira direta de determinar esse atraso é enviar pela linha um pacote especial ECHO que o outro lado deve enviar de volta imediatamente. Medindo o tempo de ida e volta e dividindo-o por dois, o roteador de envio pode obter um razoável estimativa do atraso.

Construindo pacotes de estado de link

Assim que as informações necessárias para a troca forem coletadas, o próximo A etapa é para cada roteador construir um pacote contendo todos os dados. O pacote começa com a identidade do remetente, seguido por um número de sequência e idade (a ser desriscado mais tarde) e uma lista de vizinhos. O custo para cada vizinho também é fornecido. A rede de exemplo é apresentada na Figura 5-12 (a) com os custos mostrados como rótulos na linhas. Os pacotes de link state correspondentes para todos os seis roteadores são mostrados na Figura 5-

12 (b).

B
C
E
F
UMA
D
6
1
2
8
5
7
4
3
(uma)
UMA
Seq.
Era
B
C
D
E

F
B 4
E 5
Seq.
Era
A 4
C 2
Seq.
Era
B 2
D 3
Seq.
Era
C 3
F 7
Seq.
Era
A 5
C 1
Seq.
Era
B 6
D 7
F 6
E 1
F 8
E 8
Ligaçao
Estado
Pacotes
(b)

Figura 5-12. (a) Uma rede. (b) Os pacotes de estado do link para esta rede.

Página 400

376

A CAMADA DE REDE INDIVÍDUO. 5

Construir os pacotes de estado de link é fácil. A parte difícil é determinar quando construí-los. Uma possibilidade é construí-los periodicamente, ou seja, em intervalos regulares vals. Outra possibilidade é construí-los quando algum evento significativo ocorrer, como uma linha ou vizinho caindo ou voltando novamente ou mudando seu propriedades apreciavelmente.

Distribuindo os pacotes de estado de link

A parte mais complicada do algoritmo é distribuir os pacotes de estado do link. Tudo de os roteadores devem obter todos os pacotes de estado de link de forma rápida e confiável. Se diferente

os roteadores estão usando diferentes versões da topologia, as rotas que eles calculam podem têm inconsistências, como loops, máquinas inacessíveis e outros problemas.

Primeiro, descreveremos o algoritmo de distribuição básico. Depois disso vamos dar alguns refinamentos. A ideia fundamental é usar a inundação para distribuir o conecte pacotes de estado a todos os roteadores. Para manter a inundação sob controle, cada pacote contém

um número de sequência que é incrementado para cada novo pacote enviado. Os roteadores mantêm rastrear todos os pares (roteador de origem, sequência) que eles veem. Quando um novo estado de link

o pacote chega, é verificado em relação à lista de pacotes já vista. Se for novo, ele é encaminhado em todas as linhas, exceto aquela em que chegou. Se for uma duplicata, é descartado. Se um pacote com um número de sequência menor que o maior visto assim sempre que chega, é rejeitado como obsoleto porque o roteador tem dados mais recentes.

Este algoritmo tem alguns problemas, mas são administráveis. Primeiro, se o Se os números se enrolarem, a confusão reinará. A solução aqui é usar um Número de sequência de 32 bits. Com um pacote de estado de link por segundo, levaria 137 anos para completar, então essa possibilidade pode ser ignorada. Em segundo lugar, se um roteador travar, ele perderá o controle de seu número de sequência. Se isso começar novamente em 0, o próximo pacote enviado será rejeitado como uma duplicata. Terceiro, se um número de sequência for corrompido e 65.540 for recebido de 4 (um erro de 1 bit), os pacotes de 5 a 65.540 serão rejeitados como obsoletos, uma vez que o

o número de sequência atual será considerado 65.540.

A solução para todos esses problemas é incluir a idade de cada pacote após o número da sequência e diminua-o uma vez por segundo. Quando a idade chega a zero, as informações desse roteador são descartadas. Normalmente, um novo pacote chega, digamos, a cada 10 segundos, então as informações do roteador somente expiram quando um roteador está inativo (ou seis pacotes consecutivos foram perdidos, um evento improvável). O campo *Idade* também é diminuída por cada roteador durante o processo inicial de inundação, para garantir que não pacote pode se perder e viver por um período indefinido de tempo (um pacote cuja idade é zero é descartado).

Alguns refinamentos desse algoritmo o tornam mais robusto. Quando um link state o pacote chega a um roteador para inundação, ele não é enfileirado para transmissão imediatamente. Em vez disso, é colocado em uma área de espera para esperar um pouco no caso de mais os links estão aumentando ou diminuindo. Se outro pacote de estado de link do mesmo fonte chega antes de o primeiro pacote ser transmitido, seus números de sequência são

Página 401

SEC. 5,2
ALGORITMOS DE ROTEAMENTO

377

comparado. Se eles forem iguais, a duplicata é descartada. Se eles forem diferentes, o mais velho é jogado fora. Para se proteger contra erros nos links, todos os pacotes de estado do link

são reconhecidos.

A estrutura de dados usada pelo roteador *B* para a rede mostrada na Fig. 5-12 (a) é representado na Fig. 5-13. Cada linha aqui corresponde a um recém-chegado, mas ainda não totalmente processado, pacote de estado de link. A tabela registra onde o pacote é originada, seu número de sequência e idade, e os dados. Além disso, há enviar e sinalizadores de confirmação para cada um dos três links de *B* (para *A*, *C* e *F*, respectivamente). Os sinalizadores de envio significam que o pacote deve ser enviado no ligação. Os sinalizadores de confirmação significam que ele deve ser confirmado lá.

```
D  
21  
59  
1  
0  
0  
0  
1  
1  
C  
20  
60  
1  
0  
1  
0  
1  
0  
E  
21  
59  
0  
1  
0  
1  
0  
1  
F  
21  
60  
1  
1  
0  
0  
0  
1  
UMA  
21
```

```

60
0
1
1
1
0
0
Fonte
Seq.
Era
UMA
C
F
UMA
C
F
Dados
Enviar bandeiras
Sinalizadores ACK

```

Figura 5-13. O buffer de pacotes para o roteador *B* na Figura 5.12 (a).

Na Figura 5-13, o pacote de estado do link de *A* chega diretamente, portanto, deve ser enviado para *C* e *F* e confirmados para *A*, conforme indicado pelos bits sinalizadores. Da mesma forma, o pacote et a partir de *F* tem de ser encaminhado para *um* e *C* e reconheceu a *F*.

Porém, a situação com o terceiro pacote, de *E*, é diferente. Chega duas vezes, uma vez via *EAB* e uma vez via *EFB*. Conseqüentemente, ele deve ser enviado apenas para *C*

mas deve ser confirmado por *A* e *F*, conforme indicado pelos bits.

Se uma duplicata chegar enquanto o original ainda está no buffer, os bits devem ser mudou. Por exemplo, se uma cópia do estado de *C* chegar de *F* antes do quarto entrada na tabela foi encaminhada, os seis bits serão alterados para 100011 para informe que o pacote deve ser confirmado para *F*, mas não enviado para lá.

Computando as novas rotas

Depois que um roteador acumula um conjunto completo de pacotes de estado de link, ele pode construir

todo o gráfico da rede porque cada link é representado. Cada link é, na verdade, representado duas vezes, uma para cada direção. As diferentes direções podem até têm custos diferentes. Os cálculos do caminho mais curto podem então encontrar caminhos diferentes

De um router *Um* para *B* do que de roteador *B* para *um*.

Agora o algoritmo de Dijkstra pode ser executado localmente para construir os caminhos mais curtos para

todos os destinos possíveis. Os resultados deste algoritmo informam ao roteador qual link para

Página 402

378

A CAMADA DE REDE INDIVÍDUO. 5

use para chegar a cada destino. Essas informações são instaladas nas tabelas de roteamento, e a operação normal é retomada.

Comparado ao roteamento de vetor de distância, o roteamento de estado de link requer mais memória

e computação. Para uma rede com n roteadores, cada um com k vizinhos, a memória necessária para armazenar os dados de entrada é proporcional a kn , que é pelo menos tão grande quanto uma tabela de roteamento listando todos os destinos. Além disso, o tempo de computação

cresce mais rápido do que kn , mesmo com as estruturas de dados mais eficientes, um problema em grande

redes. No entanto, em muitas situações práticas, o roteamento de link state funciona bem porque não sofre de problemas de convergência lenta.

O roteamento de estado de link é amplamente usado em redes reais, portanto, algumas palavras sobre

alguns protocolos de exemplo estão em ordem. Muitos ISPs usam o **IS-IS** (**intermediário Protocolo de estado de link System-Intermediate System**) (Oran, 1990). Foi desenhado

para uma rede antiga chamada DECnet, posteriormente adotada pela ISO para uso com o OSI protocolos e, em seguida, modificados para lidar com outros protocolos também, mais notavelmente, IP.

OSPF (Open Shortest Path First) é o outro protocolo de estado de link principal. isso foi projetado pela IETF vários anos após IS-IS e adotou muitas das inovações projetado para IS-IS. Essas inovações incluem um método autoestabilizador de inundação atualizações de estado do link, o conceito de um roteador designado em uma LAN, e o método od de computação e suporte de divisão de caminho e múltiplas métricas. Como consequência Portanto, há muito pouca diferença entre IS-IS e OSPF. O mais importante diferença significativa é que o IS-IS pode transportar informações sobre várias camadas de rede protocolos ao mesmo tempo (por exemplo, IP, IPX e AppleTalk). OSPF não tem esse recurso, e é uma vantagem em grandes ambientes multiprotocolo. Nós vamos examinar o OSPF na seção 5.6.6.

Um comentário geral sobre algoritmos de roteamento também deve ser feito. Estado do link, divisor tance, e outros algoritmos dependem do processamento em todos os roteadores para calcular rotas. Problemas com o hardware ou software mesmo em um pequeno número de roteadores pode causar estragos em toda a rede. Por exemplo, se um roteador afirma ter um link que não tem ou esquece um link que tem, o gráfico da rede será incorreta. Se um roteador não consegue encaminhar pacotes ou os corrompe durante o encaminhamento eles, a rota não funcionará conforme o esperado. Finalmente, se ficar sem memória ou faz o cálculo de roteamento errado, coisas ruins acontecerão. Conforme a rede cresce na faixa de dezenas ou centenas de milhares de nós, a probabilidade de alguns a falha do roteador ocasionalmente torna-se insignificante. O truque é tentar organizar limitar o dano quando o inevitável acontecer. Perlman (1988) discute esses problemas e suas possíveis soluções em detalhes.

5.2.6 Roteamento Hierárquico

Conforme as redes crescem em tamanho, as tabelas de roteamento do roteador aumentam proporcionalmente. Não apenas a memória do roteador é consumida por tabelas cada vez maiores, mas mais tempo de CPU é necessário para digitalizá-los e mais largura de banda é necessária para enviar relatórios de status sobre eles. Em um determinado ponto, a rede pode crescer a ponto de não ser mais

Página 403

SEC. 5,2

ALGORITMOS DE ROTEAMENTO

379

viável para cada roteador ter uma entrada para todos os outros roteadores, então o roteamento será tem que ser feito de forma hierárquica, como na rede telefônica.

Quando o roteamento hierárquico é usado, os roteadores são divididos no que iremos **regiões de** chamada . Cada roteador conhece todos os detalhes sobre como rotear pacotes para destino em sua própria região, mas não sabe nada sobre a estrutura interna de outras regiões. Quando diferentes redes estão interconectadas, é natural considerar cada um como uma região separada para liberar os roteadores em uma rede de ter que conhecer a estrutura topológica dos demais.

Para redes enormes, uma hierarquia de dois níveis pode ser insuficiente; pode ser necessário necessário agrupar as regiões em clusters, os clusters em zonas, as zonas em grupos e assim por diante, até ficarmos sem nomes para agregações. Como um exemplo de hierarquia multinível, considere como um pacote pode ser roteado de Berkeley, Califórnia fornia, para Malindi, Quênia. O roteador Berkeley saberia a topologia detalhada dentro da Califórnia, mas enviria todo o tráfego de fora do estado para o roteador de Los Angeles. O roteador de Los Angeles seria capaz de rotear o tráfego diretamente para outros roteadores, mas enviria todo o tráfego estrangeiro para Nova York. O roteador de Nova York seria programado para direcionar todo o tráfego para o roteador no país de destino responsável por lidar com o tráfego estrangeiro, digamos, em Nairóbi. Finalmente, o pacote iria

desceu da árvore no Quênia até chegar a Malindi.

A Figura 5-14 dá um exemplo quantitativo de roteamento em uma hierarquia de dois níveis com cinco regiões. A tabela de roteamento completa para o roteador *1A* tem 17 entradas, conforme mostrado em

Fig. 5-14 (b). Quando o roteamento é feito hierarquicamente, como na Figura 5-14 (c), há tenta para todos os roteadores locais, como antes, mas todas as outras regiões são condensadas em um

único roteador, de modo que todo o tráfego para a região 2 passa pela linha *1B-2A*, mas o resto do tráfego remoto passa pela linha *1C-3B*. O roteamento hierárquico reduziu a tabela de 17 a 7 entradas. Como a proporção entre o número de regiões e o número de rotas Por região cresce, a economia em espaço de mesa aumenta.

Infelizmente, esses ganhos de espaço não são gratuitos. Há uma multa a ser paga: aumento do comprimento do caminho. Por exemplo, a melhor rota de *1A* a *5C* é através da região 2, mas com o roteamento hierárquico, todo o tráfego para a região 5 passa pela região 3, porque é melhor para a maioria dos destinos na região 5.

Quando uma única rede se torna muito grande, uma questão interessante é " como muitos níveis a hierarquia deve ter? " Por exemplo, considere uma rede com 720 roteadores. Se não houver hierarquia, cada roteador precisará de 720 entradas na tabela de roteamento.

Se a rede for particionada em 24 regiões de 30 roteadores cada, cada roteador precisa 30 entradas locais mais 23 entradas remotas para um total de 53 entradas. Se um de três níveis hierarquia é escolhida, com 8 clusters cada um contendo 9 regiões de 10 roteadores, cada roteador precisa de 10 entradas para roteadores locais, 8 entradas para roteamento para outras regiões

dentro de seu próprio cluster e 7 entradas para clusters distantes, para um total de 25 entradas. Kamoun e Kleinrock (1979) descobriram que o número ideal de níveis para um

A rede do roteador N é $\ln N$, exigindo um total de $e \ln N$ entradas por roteador. Eles têm também mostrado que o aumento no comprimento do caminho médio efetivo causado pela hierarquia o roteamento é suficientemente pequeno para ser geralmente aceitável.

380

A CAMADA DE REDE INDIVÍDUO. 5

Região 1

Região 2

Região 3

Região 5

Região 4

1B

1A

1C

2A 2B

2C

5B

5C

5A

5E

5D

2D

4A

4B

4C

3A

3B

1B

1

1C

1

1B

2

1B

3

1B

3

1B

4

1C

3

```

1C
2
1C
3
1C
4
1C
4
1C
4
1C
5
1B
5
1C
6
1C
5
-
-
1A
1C
2A
2B
2C
2D
3A
3B
4A
4B
4C
5A
5B
5C
5D
5E
1B
Linha
Lúpulo
Dest.
Mesa completa para 1A
1A
1C
2
3
4
5
1B
Linha
Lúpulo
Dest.
Tabela hierárquica para 1A
1B
1
1C
1
1B
2
1C
2
1C
3
1C
4
-
-
(uma)
(b)
(c)

```

Figura 5-14. Roteamento hierárquico.

5.2.7 Roteamento de difusão

Em alguns aplicativos, os hosts precisam enviar mensagens para muitos ou todos os outros hosts. Por exemplo, um serviço de distribuição de boletins meteorológicos, atualizações do mercado de ações ou

programas de rádio podem funcionar melhor enviando para todas as máquinas e deixando que estão interessados em ler os dados. Enviar um pacote para todos os destinos simultaneamente é chamado de **transmissão**. Vários métodos foram propostos para fazer isso.

Um método de transmissão que não requer recursos especiais da rede é para a fonte simplesmente enviar um pacote distinto para cada destino. Não é só o método que desperdiça largura de banda e é lento, mas também requer que a fonte tenha uma lista completa de todos os destinos. Este método não é desejável na prática, mesmo

embora seja amplamente aplicável.

Uma melhoria é o **roteamento de vários destinos**, em que cada pacote contém uma lista de destinos ou um mapa de bits indicando os destinos desejados. Quando um pacote chega a um roteador, o roteador verifica todos os destinos para determinar o conjunto de linhas de saída que serão necessárias. (Uma linha de saída é necessária se for o melhor rota para pelo menos um dos destinos.) O roteador gera uma nova cópia do pacote para cada linha de saída a ser usado e inclui em cada pacote apenas aqueles destinos que devem usar a linha. Na verdade, o conjunto de destino é particionado entre

Página 405

SEC. 5,2

ALGORITMOS DE ROTEAMENTO

381

as linhas de saída. Depois de um número suficiente de saltos, cada pacote carregará apenas um destino como um pacote normal. O roteamento de múltiplos destinos é como usar a separação principalmente pacotes endereçados, exceto quando vários pacotes devem seguir o mesmo percurso, um deles paga a passagem inteira e o restante é grátis. A largura de banda da rede é portanto, usado de forma mais eficiente. No entanto, este esquema ainda requer a fonte para saber todos os destinos, além disso, é muito trabalhoso para um roteador determinar onde para enviar um pacote com vários destinos, pois é para vários pacotes distintos.

Já vimos uma técnica de roteamento de broadcast melhor: flooding. Quando implementado com um número de sequência por fonte, o flooding usa links de forma eficiente com uma regra de decisão em roteadores que é relativamente simples. Embora a inundação seja prejudicial

adequado para comunicação ponto-a-ponto comum, ele avalia seriamente transmissão. No entanto, descobrimos que podemos fazer melhor ainda, uma vez que o mais curto rotas de caminho para pacotes regulares foram calculadas.

A ideia de **encaminhamento de caminho reverso** é elegante e extremamente simples uma vez foi apontado (Dalal e Metcalfe, 1978). Quando um pacote de broadcast é rives em um roteador, o roteador verifica se o pacote chegou ao link que está normalmente usado para enviar pacotes *para* a origem da transmissão. Se sim, lá é uma excelente chance de que o próprio pacote de transmissão tenha seguido a melhor rota de o roteador e, portanto, é a primeira cópia a chegar ao roteador. Sendo este o Nesse caso, o roteador encaminha cópias dele para todos os links, exceto aquele em que chegou. E se,

no entanto, o pacote de transmissão chegou em um link diferente do preferido para ao chegar à origem, o pacote é descartado como uma provável duplicata.

Eu
F
H
J
N
UMA
D
G
K
E
O
M
O
G
C
D
N
B
H
eu
eu
B
UMA
E
H
B
C
D
F
J
G
O
M

K
eu
N
Eu
(uma)
UMA
B
C
D
G
J
O
F
Eu
E
H
K
eu
M
N
(b)
(c)
K
E
H

Figura 5-15. Encaminhamento de caminho reverso. (a) Uma rede. (b) Uma árvore que afunda. (c) O árvore construída por encaminhamento de caminho reverso.

Um exemplo de encaminhamento de caminho reverso é mostrado na Figura 5-15. Parte (a) mostra uma rede, a parte (b) mostra uma árvore de coletor para o roteador *I* dessa rede, e a parte (c) mostra como funciona o algoritmo de caminho reverso. No primeiro hop, *eu* envia pacotes para *F*, *H*, *J* e *N*, conforme indicado pela segunda linha da árvore. Cada um desses pacotes chega no caminho preferido para *I* (assumindo que o caminho preferido cai ao longo do afundar) e é indicado por um círculo ao redor da letra. No segundo salto,

Página 406

382

A CAMADA DE REDE INDIVÍDUO. 5

oito pacotes são gerados, dois por cada um dos roteadores que receberam um pacote em o primeiro salto. Acontece que todos os oito chegam a uma rota não visitada anteriormente e cinco deles chegam ao longo da linha preferida. Dos seis pacotes gerados no terceiro salto, apenas três chegam ao caminho preferido (em *C*, *E* e *K*); o outros são duplicados. Depois de cinco saltos e 24 pacotes, a transmissão termina, em comparação com quatro saltos e 14 pacotes, a árvore coletor foi seguida exatamente.

A principal vantagem do encaminhamento de caminho reverso é que ele é eficiente enquanto sendo fácil de implementar. Ele envia o pacote de broadcast por cada link apenas uma vez em cada direção, assim como na inundação, mas requer apenas que os roteadores saibam como chegar a todos os destinos, sem a necessidade de lembrar os números de sequência (ou usar outros mecanismos para interromper a inundação) ou listar todos os destinos no pacote.

Nosso último algoritmo de transmissão melhora o comportamento do caminho reverso para guarda. Ele faz uso explícito da árvore coletor - ou qualquer outra extensão conveniente árvore - para o roteador que inicia a transmissão. Uma **árvore de abrangência** é um subconjunto do rede que inclui todos os roteadores, mas não contém loops. Árvores de coletor estão abrangendo árvores. Se cada roteador souber quais de suas linhas pertencem à árvore de abrangência, ele pode copiar um pacote de transmissão de entrada em todas as linhas da árvore de abrangência, exceto aquela

chegou. Este método faz um excelente uso da largura de banda, gerando o número mínimo absoluto de pacotes necessários para fazer o trabalho. Na Fig. 5-15, para exemplo, quando o sink tree da parte (b) é usado como spanning tree, o broadcast o pacote é enviado com no mínimo 14 pacotes. O único problema é que cada roteador deve ter conhecimento de alguma árvore abrangente para que o método seja aplicável.

Às vezes, essa informação está disponível (por exemplo, com roteamento de estado de link, todos os roteadores

conhecem a topologia completa, para que possam calcular uma árvore de abrangência), mas às vezes não é (por exemplo, com roteamento de vetor de distância).

5.2.8 Roteamento Multicast

Alguns aplicativos, como um jogo multijogador ou vídeo ao vivo de um evento esportivo transmitido para muitos locais de visualização, envie pacotes para vários receptores. A menos que o grupo é muito pequeno, enviar um pacote distinto para cada receptor é caro. Por outro lado, transmitir um pacote é um desperdício se o grupo consiste em, digamos, 1000 máquinas em uma rede de um milhão de nós, para que a maioria dos receptores não sejam interessados na mensagem (ou pior ainda, eles estão definitivamente interessados, mas não são posso ver). Assim, precisamos de uma maneira de enviar mensagens para grupos bem definidos que são numericamente grandes em tamanho, mas pequenos em comparação com a rede como um todo.

O envio de uma mensagem para esse grupo é chamado de **multicast**, e o roteamento também O goritmo usado é chamado de **roteamento multicast**. Todos os esquemas de multicast requerem algum

maneira de criar e destruir grupos e identificar quais roteadores são membros de um grupo. Como essas tarefas são realizadas não é uma preocupação para o algoritmo de roteamento ritmo. Por enquanto, vamos assumir que cada grupo é identificado por um anúncio multicast vestido e que os roteadores saibam os grupos a que pertencem. Vamos revisitar membros do grupo quando descrevemos a camada de rede da Internet em seg. 5,6.

Página 407

SEC. 5,2
ALGORITMOS DE ROTEAMENTO

383

Os esquemas de roteamento multicast constroem nos esquemas de roteamento de broadcast que al- pronto estudo, enviando pacotes ao longo de árvores abrangentes para entregar os pacotes ao membros do grupo enquanto fazem uso eficiente da largura de banda. No entanto, o a melhor árvore de abrangência a usar depende se o grupo é denso, com receptores espalhadas pela maior parte da rede, ou esparsas, com grande parte da rede não sendo saudade do grupo. Nesta seção, consideraremos os dois casos.

Se o grupo for denso, a transmissão é um bom começo porque obtém com eficiência o pacote para todas as partes da rede. Mas a transmissão alcançará alguns roteadores que são não membros do grupo, o que é um desperdício. A solução explorada por Deering e Cheriton (1990) é podar a árvore de abrangência da transmissão removendo links que não conduza a membros. O resultado é uma árvore de abrangência multicast eficiente.

Como exemplo, considere os dois grupos, 1 e 2, na rede mostrada em

Fig. 5-16 (a). Alguns roteadores são anexados a hosts que pertencem a um ou a ambos esses grupos, conforme indicado na figura. Uma árvore estendida para o roteador mais à esquerda é mostrado na Fig. 5-16 (b). Esta árvore pode ser usada para transmissão, mas é um exagero para mu- licast, como pode ser visto nas duas versões podadas que são mostradas a seguir. No Fig. 5-16 (c), todos os links que não levam a hosts que são membros do grupo 1 foi removido. O resultado é a árvore de abrangência multicast para o mais à esquerda roteador para enviar ao grupo 1. Os pacotes são encaminhados apenas ao longo desta árvore de abrangência,

que é mais eficiente do que a árvore de transmissão porque existem 7 links em vez de 10. A Fig. 5-16 (d) mostra a árvore de abrangência multicast após a poda para o grupo 2. É eficiente também, com apenas cinco links desta vez. Também mostra que diferentes multicast grupos têm diferentes árvores abrangentes.

São possíveis várias maneiras de podar a árvore geradora. O mais simples pode ser usado se o roteamento de estado de link for usado e cada roteador estiver cliente do topo completo logia, incluindo quais hosts pertencem a quais grupos. Cada roteador pode então controlar estruturar sua própria árvore de abrangência podada para cada remetente para o grupo em questão por

construir uma árvore coletor para o remetente como de costume e, em seguida, remover todos os links que

não conectar membros do grupo ao nó coletor. **MOSPF (Multicast OSPF)** é um exemplo de um protocolo de estado de link que funciona dessa forma (Moy, 1994).

Com o roteamento do vetor de distância, uma estratégia de poda diferente pode ser seguida.

O algoritmo básico é o encaminhamento de caminho reverso. No entanto, sempre que um roteador com nenhum host interessado em um determinado grupo e nenhuma conexão com outros roteadores recebe uma mensagem multicast para esse grupo, ele responde com uma mensagem PRUNE , tell-o vizinho que enviou a mensagem para não enviar mais nenhum multicasts do remetente para esse grupo. Quando um roteador sem membros de grupo entre seus próprios hosts recebeu essas mensagens em todas as linhas para as quais envia o multicast, também pode responder com uma mensagem PRUNE . Desta forma, a árvore de abrangência é recursivamente podado. **DVMRP (Distance Vector Multicast Routing Protocol)** é um exemplo de um protocolo de roteamento multicast que funciona dessa forma (Waitzman et al., 1988). A poda resulta em árvores geradoras eficientes que usam apenas os links que estão ativos necessário para alcançar os membros do grupo. Uma desvantagem potencial é que é muito trabalhoso para roteadores, especialmente para grandes redes. Suponha que uma rede

Página 408

384

A CAMADA DE REDE

INDIVÍDUO. 5

1, 2

1

1, 2

2

1

1

2

2

1

2

1, 2

1, 2

2

2

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

A Fig. 5-17 (a) mostra uma árvore baseada em núcleo para o grupo 1. Para enviar a este grupo, um remetente envia um pacote para o núcleo. Quando o pacote atinge o núcleo, ele é encaminhado para baixo a árvore. Isso é mostrado na Fig. 5-17 (b) para o remetente no lado direito do rede. Como uma otimização de desempenho, os pacotes destinados ao grupo não precisam alcançar o núcleo antes de serem multicast. Assim que um pacote chega ao

Página 409

SEC. 5,2
ALGORITMOS DE ROTEAMENTO

385

árvore, pode ser encaminhado para cima em direção à raiz, bem como para baixo em todas as outras ramos. Esse é o caso do remetente da Figura 5-17 (b).

```
1
1
1
1
1
1
1
1
1
1
Testemunho
Testemunho
Remetente
Remetente
(uma)
(b)
```

Figura 5-17. (a) Árvore baseada em núcleo para o grupo 1. (b) Enviando para o grupo 1.

Ter uma árvore compartilhada não é o ideal para todas as fontes. Por exemplo, na Fig. 5-17 (b), o pacote do remetente do lado direito atinge o grupo superior direito membro através do núcleo em três saltos, em vez de diretamente. A ineficiência depende sobre onde o núcleo e os remetentes estão localizados, mas muitas vezes é razoável quando o núcleo está no meio dos remetentes. Quando há apenas um remetente, como em um vídeo que é transmitido para um grupo, usando o remetente como o núcleo é ideal.

Também digno de nota é que as árvores compartilhadas podem ser uma grande economia nos custos de armazenamento, mes-

sábios enviados e computação. Cada roteador deve manter apenas uma árvore por grupo, em vez de árvores m . Além disso, os roteadores que não fazem parte da árvore não funcionam para apoiar o grupo. Por esta razão, abordagens de árvore compartilhada como árvores baseadas em núcleo

são usados para multicast para grupos esparsos na Internet como parte de um protocolo popular cols como **PIM** (Protocol Independent Multicast) (Fenner et al., 2006).

5.2.9 Roteamento Anycast

Até agora, cobrimos os modelos de entrega em que uma fonte envia para um único destino (chamado de **unicast**), para todos os destinos (chamado de transmissão) e para um grupo de destinos (chamado multicast). Outro modelo de entrega, chamado **anycast** é às vezes também útil. Em anycast, um pacote é entregue ao membro mais próximo de um grupo (Partridge et al., 1993). Os esquemas que encontram esses caminhos são chamados de **anycast roteamento** .

Por que queremos anycast? Às vezes, os nós fornecem um serviço, como hora do dia ou distribuição de conteúdo para a qual está obtendo as informações certas isso importa, não o nó que é contatado; qualquer nó servirá. Por exemplo, qualquer O elenco é usado na Internet como parte do DNS, como veremos no cap. 7 Felizmente, não teremos que criar novos esquemas de roteamento para anycast porque o vetor de distância regular e o roteamento de estado de link podem produzir rotas anycast. Suponha

Página 410

386

A CAMADA DE REDE

INDIVÍDUO. 5

queremos transmitir para os membros do grupo 1. Todos receberão o endereço "1," em vez de endereços diferentes. O roteamento do vetor de distância distribuirá os vetores como de costume, e os nós escolherão o caminho mais curto para o destino 1. Isso resultará em nós enviando para a instância mais próxima de destino 1. As rotas são mostradas em Fig. 5-18 (a). Este procedimento funciona porque o protocolo de roteamento não realiza que existem várias instâncias de destino 1. Ou seja, ele acredita que todos os as instâncias do nó 1 são o mesmo nó, como na topologia mostrada na Figura 5.18 (b).

1
1
1
1
1
1
(uma)
(b)

Figura 5-18. (a) Rotas anycast para o grupo 1. (b) Topologia vista pelo protocolo de roteamento.

Este procedimento funciona para o roteamento de estado de link também, embora haja a consideração adicional de que o protocolo de roteamento não deve encontrar caminhos aparentemente curtos

que passam pelo nó 1. Isso resultaria em saltos pelo hiperespaço, uma vez que as instâncias do nó 1 são, na verdade, nós localizados em diferentes partes da rede.

No entanto, os protocolos de estado de link já fazem essa distinção entre roteadores e hospedeiros. Nós encobrimos esse fato antes porque não era necessário para nossa doença cussão.

5.2.10 Roteamento para hosts móveis

Milhões de pessoas usam computadores em trânsito, em situações verdadeiramente móveis ções com dispositivos sem fio em carros em movimento, para situações nômades em que os computadores são usados em uma série de locais diferentes. Usaremos o termo **móvel hosts** para significar qualquer uma das categorias, diferentemente de hosts estacionários que nunca se movem.

Cada vez mais, as pessoas querem ficar conectadas em qualquer lugar do mundo em que estejam, pois

facilmente como se estivessem em casa. Esses hosts móveis apresentam uma nova complicação: para rotear um pacote para um host móvel, a rede primeiro precisa localizá-lo.

O modelo de mundo que consideraremos é aquele em que todos os hospedeiros são como somava ter um **local de residência** permanente que nunca muda. Cada hospedeiro também tem um endereço residencial permanente que pode ser usado para determinar seu local de residência, análogo ao modo como o número de telefone 1-212-5551212 indica a Estados (código de país 1) e Manhattan (212). O objetivo de roteamento em sistemas com

Página 411

SEC. 5,2

ALGORITMOS DE ROTEAMENTO

387

hosts móveis é tornar possível o envio de pacotes para hosts móveis usando seus endereços residenciais fixos e fazer com que os pacotes cheguem com eficiência onde quer que eles talvez. O truque, é claro, é encontrá-los.

Alguma discussão deste modelo é necessária. Um modelo diferente seria recomputar as rotas conforme o host móvel se move e a topologia muda. Poderíamos em seguida, basta usar os esquemas de roteamento descritos anteriormente nesta seção. Contudo, com um número crescente de hosts móveis, esse modelo logo levaria a todo o rede computando continuamente novas rotas. Usando os endereços residenciais muito reduz esse fardo.

Outra alternativa seria fornecer mobilidade acima da camada de rede, que é o que normalmente acontece com laptops hoje. Quando eles são movidos para o novo Locais da Internet, laptops adquirem novos endereços de rede. Não há associação entre o antigo e o novo endereço; a rede não sabe que eles pertenciam para o mesmo laptop. Neste modelo, um laptop pode ser usado para navegar na web, mas outros hosts não podem enviar pacotes para ele (por exemplo, para uma chamada recebida), sem

construir um serviço de localização de camada superior, por exemplo, entrar no Skype *novamente* depois de se mover. Além disso, as conexões não podem ser mantidas enquanto o host está movido; em vez disso, novas conexões devem ser iniciadas. A mobilidade da camada de rede é útil para consertar esses problemas.

A ideia básica usada para roteamento móvel na Internet e redes celulares é para o host móvel informar a um host na localização inicial onde ele está agora. Este hospedeiro, que atua em nome do host móvel, é chamado de **agente doméstico**. Uma vez que sabe onde o host móvel está localizado, ele pode encaminhar pacotes para que eles sejam entregue.

A Figura 5-19 mostra o roteamento móvel em ação. Um remetente na cidade noroeste de Seattle deseja enviar um pacote a um host normalmente localizado nos Estados Unidos Em Nova Iorque. O caso que nos interessa é quando o host móvel não está em casa. Em vez disso, está temporariamente em San Diego.

O host móvel em San Diego deve adquirir um endereço de rede local antes de pode usar a rede. Isso acontece da maneira normal que os hosts obtêm rede endereços; vamos cobrir como isso funciona para a Internet mais adiante neste capítulo. o o endereço local é denominado **endereço de cuidado**. Assim que o host móvel tiver esse endereço, ele pode dizer ao seu agente local onde está agora. Ele faz isso enviando um registro mensagem para o agente doméstico (etapa 1) com o cuidado do endereço. A mensagem é mostrado com uma linha tracejada na Fig. 5-19 para indicar que é uma mensagem de controle, não uma mensagem de dados.

Em seguida, o remetente envia um pacote de dados para o host móvel usando seu endereço (etapa 2). Este pacote é roteado pela rede para a localização inicial do host porque é onde o endereço residencial pertence. Em Nova York, o agente doméstico intercepta este pacote porque o host móvel está longe de casa. Em seguida, envolve ou **encapsula** o pacote com um novo cabeçalho e envia este pacote aos cuidados de endereço (etapa 3). Esse mecanismo é chamado de **tunelamento**. É muito importante no Internet, portanto, veremos com mais detalhes posteriormente.

Página 412

388

A CAMADA DE REDE

INDIVÍDUO. 5

Host móvel em

cuidar do endereço

3: Túnel para cuidar do endereço

1: Registercare of address

2: Send tohomeaddress

Agente doméstico em

endereço residencial

Remetente

4: Responder

ao remetente

5: Túnel

cuidar de

endereço

Figura 5-19. Roteamento de pacotes para hosts móveis.

Quando o pacote encapsulado chega aos cuidados do endereço, o host móvel desembrulha-o e recupera o pacote do remetente. O host móvel então envia seu pacote de resposta diretamente para o remetente (etapa 4). A rota geral é chamada de **triângulo roteamento** porque pode ser tortuoso se o local remoto for longe de casa localização. Como parte da etapa 4, o remetente pode aprender o endereço atual. Sub-pacotes sequentes podem ser roteados diretamente para o host móvel, tunelando-os para o cuidar do endereço (etapa 5), ignorando totalmente o local de residência. Se a conectividade for perdido por qualquer motivo enquanto o celular se move, o endereço residencial sempre pode ser usado para chegar ao celular.

Um aspecto importante que omitimos desta descrição é a segurança. No geral, quando um host ou roteador recebe uma mensagem no formato " Começando agora, por favor, envie todos os e-mails de Stephany para mim, " pode haver algumas perguntas

sobre com quem está falando e se isso é uma boa ideia. Informação segura está incluído nas mensagens para que sua validade possa ser verificada com criptografia protocolos gráficos que estudaremos no cap. 8

Existem muitas variações no roteamento móvel. O esquema acima é modelado sobre a mobilidade IPv6, a forma de mobilidade usada na Internet (Johnson et al., 2004) e como parte de redes celulares baseadas em IP, como UMTS. Mostramos ao remetente ser um nó estacionário para simplificar, mas os projetos permitem que ambos os nós sejam móveis hospedeiros. Alternativamente, o host pode ser parte de uma rede móvel, por exemplo, um computador em um avião. Extensões do esquema básico suportam redes móveis sem trabalho por parte dos hospedeiros (Devarapalli et al., 2005).

Alguns esquemas fazem uso de um agente estrangeiro (ou seja, remoto), semelhante ao da casa agente, mas em local estrangeiro, ou análogo ao VLR (Visitor Location Register) em redes celulares. No entanto, em esquemas mais recentes, o agente estrangeiro é não é necessário; hosts móveis atuam como seus próprios agentes estrangeiros. Em qualquer caso, saiba

borda da localização temporária do host móvel é limitada a um pequeno número de

Página 413

SEC. 5,2
ALGORITMOS DE ROTEAMENTO

389

hosts (por exemplo, o celular, o agente doméstico e os remetentes) para os muitos roteadores em um grandes redes não precisam recalcular as rotas.

Para obter mais informações sobre o roteamento móvel, consulte também Perkins (1998, 2002) e Snoeren e Balakrishnan (2000).

5.2.11 Roteamento em Redes Ad Hoc

Agora vimos como fazer o roteamento quando os hosts são móveis, mas o roteamento é fixo. Um caso ainda mais extremo é aquele em que os próprios roteadores são móveis. Entre as possibilidades estão os trabalhadores de emergência em um local do terremoto, veículos militares em um campo de batalha, uma frota de navios no mar ou uma reunião de pessoas com laptops em uma área sem 802.11.

Em todos esses casos, e em outros, cada nó se comunica sem fio e atua como um host e um roteador. Redes de nós que estão próximos uns dos outros são chamadas de **redes ad hoc** ou **MANETs (Mobile Ad hoc NETworks)**. Deixe-nos agora examinar brevemente. Mais informações podem ser encontradas em Perkins (2001). O que torna as redes ad hoc diferentes das redes com fio é que o topo logy é repentinamente atirado pela janela. Os nós podem ir e vir ou aparecer em novos lugares na queda de um pouco. Com uma rede com fio, se um roteador tem um caminho válido para algum destino, esse caminho continua a ser falhas de barramento válidas, que são esperança-totalmente raro. Com uma rede ad hoc, a topologia pode mudar o tempo todo, então a desejabilidade e até mesmo a validade dos caminhos podem mudar espontaneamente sem atenção. Desnecessário dizer que essas circunstâncias tornam o roteamento em redes ad hoc mais desafiador do que o roteamento em suas contrapartes fixas.

Muitos, muitos algoritmos de roteamento para redes ad hoc foram propostos. No entanto, uma vez que as redes ad hoc têm sido pouco utilizadas na prática em comparação com redes móveis, não está claro quais desses protocolos são mais úteis. Como exemplificaremos, veremos um dos algoritmos de roteamento mais populares, **AODV (Ad hoc On-demand Distance Vector)** (Perkins e Royer, 1999). É parente de

o algoritmo do vetor de distância que foi adaptado para funcionar em um ambiente móvel, em que os nós geralmente têm largura de banda limitada e vida útil da bateria. Deixe-nos agora ver como ele descobre e mantém as rotas.

Descoberta de rota

No AODV, as rotas para um destino são descobertas sob demanda, ou seja, apenas quando alguém deseja enviar um pacote para aquele destino. Isso economiza muito trabalho que de outra forma seria desperdiçado quando a topologia muda antes da rota é usado. A qualquer momento, a topologia de uma rede ad hoc pode ser descrita por um

gráfico de nós conectados. Dois nós estão conectados (ou seja, têm um arco entre no gráfico) se eles podem se comunicar diretamente usando seus rádios. Um básico mas o modelo adequado que é suficiente para nossos propósitos é que cada nó pode compor comunicar-se com todos os outros nós que estão dentro de seu círculo de cobertura. Redes reais são

390

A CAMADA DE REDE

INDIVÍDUO. 5

mais complicado, com edifícios, colinas e outros obstáculos que bloqueiam a comunicação, e nós para os quais A está conectado a B , mas B não está conectado a A - causa Um tem um transmissor mais potente do que B . No entanto, para simplificar, vamos assumir que todas as conexões são simétricas.

Para descrever o algoritmo, considere a rede ad hoc recém-formada de

Fig. 5-20. Suponha que um processo no nó A quer enviar um pacote ao nó I . O algoritmo AODV mantém uma tabela de vetor de distância em cada nó, codificado por destino, dando informações sobre esse destino, incluindo o vizinho ao qual para enviar pacotes para chegar ao destino. Primeiro, A olha em sua mesa e não encontrar uma entrada para I . Agora tem que descobrir uma rota para eu . Esta propriedade de descobrir

carregar rotas apenas quando elas são necessárias é o que torna esse algoritmo " sob demanda ".

UMA

B

C

Alcance de

Transmissão de A

UMA

D

B

C

E

F

H

Eu

H

Eu

G

G

E

E

D

D

C

C

B

B

UMA

UMA

G

Eu

H

F

D

E

G

Eu

H

(uma)

(b)

(c)

(d)

F

Figura 5-20. (a) Faixa de transmissão de A . (b) Depois de B e D recebê-lo. (c) Depois de C , F e G recebem. (d) Depois de E , H e eu recebem. Os nós sombreados são novos destinatários. As linhas tracejadas mostram possíveis rotas reversas. As linhas sólidas mostrar a rota descoberta.

Para localizar I , A constrói um pacote ROUTE REQUEST e o transmite usando inundação, conforme descrito na Seç. 5.2.3. A transmissão de A atinge B e D , conforme ilustrado na Fig. 5-20 (a). Cada nó retransmite a solicitação, que continua a alcançar os nós F , G e C na Fig. 5-20 (c) e os nós H , E e I na Fig. 5-20 (d). UMA o número de sequência definido na fonte é usado para eliminar duplicatas durante a enchente.

Por exemplo, D descarta a transmissão de B na Fig. 5-20 (c) porque tem tudo pronto encaminhou o pedido.

Eventualmente, a solicitação chega ao nó I , que constrói um ROUTE REPLY pacote. Este pacote é unicast para o remetente ao longo do caminho inverso seguido pelo pedido. Para que isso funcione, cada nó intermediário deve se lembrar do nó que enviou a solicitação. As setas na Fig. 5-20 (b) - (d) mostram a rota reversa informações que são armazenadas. Cada nó intermediário também incrementa uma contagem de saltos como ele encaminha a resposta. Isso informa aos nós a que distância eles estão do destino. As respostas dizem a cada nó intermediário qual vizinho usar para alcançar o destino: é o nó que lhes enviou a resposta. Os nós intermediários G e D colocam o

Página 415

SEC. 5,2 ALGORITMOS DE ROTEAMENTO

391

melhor rota que ouvem em suas tabelas de roteamento à medida que processam a resposta. Quando a resposta chega a A , uma nova rota, $ADGI$, foi criada.

Em uma grande rede, o algoritmo gera muitas transmissões, mesmo para destinos próximos que estão por perto. Para reduzir a sobrecarga, o escopo das transmissões é limitado usando o campo *Time to live* do pacote IP. Este campo é inicializado pelo remetente e diminuído em cada salto. Se chegar a 0, o pacote é descartado em vez de ser transmitido. O processo de descoberta de rota é então modificado da seguinte maneira. Para localizar um destino, o remetente transmite um pacote ROUTE REQUEST com *Time to live* definido a 1. Se nenhuma resposta retornar dentro de um tempo razoável, outra é enviada, esta tempo com o *tempo de vida* definido para 2. As tentativas subsequentes usam 3, 4, 5, etc. Desta forma,

a busca é tentada primeiro localmente, depois em anéis cada vez mais amplos.

Manutenção de rota

Como os nós podem se mover ou ser desligados, a topologia pode mudar instantaneamente. Por exemplo, na Fig. 5-20, se G estiver desligado, A não perceberá que a rota que estava usando para I ($ADGI$) não é mais válida. O algoritmo precisa ser capaz de lidar com isso. Periodicamente, cada nó transmite uma mensagem de *saudação*. Cada espera-se que seus vizinhos respondam a ela. Se não houver resposta, o

a emissora sabe que aquele vizinho saiu do alcance ou falhou e não mais conectado a ele. Da mesma forma, se ele tenta enviar um pacote para um vizinho que não responde, fica sabendo que o vizinho não está mais disponível.

Essas informações são usadas para limpar as rotas que não funcionam mais. Para cada destino possível, cada nó, N , mantém o controle de seus vizinhos ativos que o alimentaram um pacote para esse destino durante os últimos ΔT segundos. Quando qualquer um de N 's neighbors se torna inacessível, ele verifica sua tabela de roteamento para ver quais destinos têm rotas usando o vizinho que agora se foi. Para cada uma dessas rotas, o ativo os vizinhos são informados de que sua rota via N agora é inválida e deve ser eliminada de suas tabelas de roteamento. Em nosso exemplo, D limpa suas entradas para G e I de seus encaminhamento mesa e notifica A , o que elimina a sua entrada para I . No caso geral, os vizinhos ativos contam a seus vizinhos ativos, e assim por diante, recursivamente, até que todas as rotas

dependendo do nó desaparecido, são removidos de todas as tabelas de roteamento.

Nesta fase, as rotas inválidas foram eliminadas da rede e o envio-ers podem encontrar novas rotas válidas usando o mecanismo de descoberta que derabiscado. No entanto, existe uma complicação. Lembre-se de que protocolos de vetor de distância pode sofrer de convergência lenta ou problemas de contagem até o infinito após uma topologia mudança na qual eles confundem rotas antigas e inválidas com rotas novas e válidas.

Para garantir uma convergência rápida, as rotas incluem um número de sequência que é conseqüentemente pelo destino. O número de sequência de destino é como uma lógica relógio. O destino o incrementa toda vez que envia uma nova ROTA

REPLY . Os remetentes pedem uma nova rota, incluindo em ROUTE REQUEST o número de sequência de destino da última rota que usaram, que será o número de sequência da rota que acabou de ser eliminada ou 0 como um valor inicial. o

392

A CAMADA DE REDE

INDIVÍDUO. 5

a solicitação será transmitida até que uma rota com um número de sequência mais alto seja encontrada.

Nós intermediários armazenam as rotas que têm um número de sequência mais alto, ou o menor número de saltos para o número de sequência atual.

No espírito de um protocolo sob demanda, os nós intermediários armazenam apenas as rotas que estão em uso. Outras informações de rota aprendidas durante as transmissões são cronometradas

após um pequeno atraso. Descobrir e armazenar apenas as rotas que são usadas ajuda para economizar largura de banda e vida útil da bateria em comparação com um protocolo de vetor de distância padrão

que transmite atualizações periodicamente.

Até agora, temos considerado apenas uma única rota, de A a I . Para salvar ainda mais recursos, descoberta de rota e manutenção são compartilhados quando as rotas se sobreponem. Para Por exemplo, se B também quiser enviar pacotes para I , ele executará a descoberta da rota.

No entanto, neste caso, o pedido será primeiro chegar D , que já tem uma rota para I .

O Nó D pode então gerar uma resposta para dizer a B a rota sem nenhum trabalho adicional sendo necessário.

Existem muitos outros esquemas de roteamento ad hoc. Outro conhecido em de- O esquema de mand é DSR (Dynamic Source Routing) (Johnson et al., 2001). A dif- estratégia diferente baseada na geografia é explorada por GPSR (Greedy Perimeter State- menos Routing) (Karp e Kung, 2000). Se todos os nós souberem suas posições geográficas ções, o encaminhamento para um destino pode prosseguir sem o cálculo da rota por simulação

ply indo na direção certa e circulando de volta para escapar de quaisquer becos sem saída. Quais protocolos vencerão dependerá dos tipos de redes ad hoc que comprovam útil na prática.

5.3 ALGORITMOS DE CONTROLE DE CONGESTÃO

Muitos pacotes presentes (em uma parte) da rede causam atrasos nos pacotes e perda que degrada o desempenho. Essa situação é chamada de **congestionamento** . A rede e as camadas de transporte compartilham a responsabilidade de lidar com o congestionamento. Desde con-

gestão ocorre dentro da rede, é a camada de rede que experimenta diretamente e deve determinar o que fazer com os pacotes em excesso. Contudo, a maneira mais eficaz de controlar o congestionamento é reduzir a carga que o transporte camada de esporte está sendo colocado na rede. Isso requer a rede e a configuração de transporte funcionários para trabalharem juntos. Neste capítulo, veremos os aspectos da rede de congestão. No cap. 6, concluiremos o tópico cobrindo os aspectos de transporte de congestionamento.

A Figura 5-21 mostra o início do congestionamento. Quando o número de pacotes hosts enviados para a rede estão dentro de sua capacidade de transporte, o número entregue é proporcional ao número enviado. Se o dobro for enviado, o dobro são entregues. No entanto, à medida que a carga oferecida se aproxima da capacidade de carga, rajadas de tráfego ocasionalmente enchem os buffers dentro dos roteadores e alguns pacotes estão perdidos. Esses pacotes perdidos consomem parte da capacidade, portanto, o número de os pacotes entregues ficam abaixo da curva ideal. A rede agora está congestionada.

SEC. 5,3

ALGORITMOS DE CONTROLE DE CONGESTÃO

393

Ideal
Goodput
(pacotes / s)
Desejável
resposta
Capacidade de
a rede
Congestionamento
colapso
Carga oferecida (pacote / s)
Início de
congestionamento

Figura 5-21. Com muito tráfego, o desempenho cai drasticamente.

A menos que a rede seja bem projetada, pode ocorrer um **colapso do congestionamento**, em que o desempenho cai conforme a carga oferecida aumenta além da capacidade. Isso pode acontecer porque os pacotes podem ser suficientemente atrasados dentro da rede trabalho que eles não são mais úteis quando saem da rede. Por exemplo, em o início da Internet, o tempo que um pacote passava esperando por um acúmulo de pacotes à frente dele para ser enviado por um link lento de 56 kbps poderia atingir o tempo máximo que já era reduzidos a permanecer na rede. Em seguida, teve que ser jogado fora. Uma falha diferente modo ocorre quando os remetentes retransmitem pacotes que estão muito atrasados, pensando que eles foram perdidos. Neste caso, cópias do mesmo pacote serão entregues pela rede, mais uma vez desperdiçando sua capacidade. Para capturar esses fatores, o eixo y de A Fig. 5-21 é dada como **goodput**, que é a taxa na qual os pacotes *úteis* são entregues gerada pela rede.

Gostaríamos de projetar redes que evitem o congestionamento sempre que possível e não sofra de colapso de congestionamento se eles ficarem congestionados. Infelizmente, o congestionamento não pode ser totalmente evitado. De repente, fluxos de pacotes começam a chegar em três ou quatro linhas de entrada e todas precisam da mesma linha de saída, um a fila vai aumentar. Se não houver memória suficiente para armazenar todos eles, os pacotes será perdido. Adicionar mais memória pode ajudar até certo ponto, mas Nagle (1987) real percebeu que, se os roteadores têm uma quantidade infinita de memória, o congestionamento fica pior, não

Melhor. Isso ocorre porque quando os pacotes chegam à frente da fila, eles têm já expirou (repetidamente) e duplicatas foram enviadas. Isso torna importante pior, não melhor - leva ao colapso do congestionamento.

Links ou roteadores de baixa largura de banda que processam pacotes mais lentamente do que a linha

a taxa também pode ficar congestionada. Neste caso, a situação pode ser melhorada direcionar parte do tráfego do gargalo para outras partes da rede trabalhos. Eventualmente, no entanto, todas as regiões da rede ficarão congestionadas. Nisso situação, não há alternativa a não ser reduzir a carga ou construir uma rede mais rápida.

Vale ressaltar a diferença entre controle de congestionamento e fluxo controle, já que o relacionamento é muito sutil. O controle de congestionamento tem a ver com

Página 418

394

A CAMADA DE REDE
INDIVÍDUO. 5

certificando-se de que a rede é capaz de transportar o tráfego oferecido. É um problema global, em envolvendo o comportamento de todos os hosts e roteadores. O controle de fluxo, em contraste, está relacionado

ao tráfego entre um determinado remetente e um determinado receptor. Seu trabalho é certifique-se de que um remetente rápido não pode transmitir dados continuamente mais rápido do que o receptor é capaz de absorvê-lo.

Para ver a diferença entre esses dois conceitos, considere uma rede feita de links de fibra óptica de 100 Gbps nos quais um supercomputador está tentando forçar a alimentação

um arquivo grande para um computador pessoal que é capaz de lidar com apenas 1 Gbps. Al-

embora não haja congestionamento (a própria rede não está com problemas), o controle de fluxo está precisava forçar o supercomputador a parar com frequência para dar a comunicação pessoal puter uma chance de respirar.

No outro extremo, considere uma rede com linhas de 1 Mbps e 1000 grandes computadores, metade dos quais tentando transferir arquivos a 100 kbps para a outra metade. Aqui, o problema não é o de remetentes rápidos sobrepujando os receptores lentos, mas que o tráfego total oferecido excede o que a rede pode suportar.

A razão pela qual o controle de congestionamento e o controle de fluxo são frequentemente confundidos é que o

a melhor maneira de lidar com os dois problemas é desacelerar o host. Assim, um hospedeiro pode receber uma mensagem de "desaceleração" porque o receptor não consegue lidar com o carregar ou porque a rede não consegue lidar com isso. Voltaremos a este ponto em Indivíduo. 6

Começaremos nosso estudo de controle de congestionamento observando as abordagens que pode ser usado em diferentes escalas de tempo. Em seguida, veremos abordagens para pré-evitando que o congestionamento ocorra em primeiro lugar, seguido por abordagens para lidar com isso, uma vez que tenha estabelecido.

5.3.1 Abordagens para controle de congestionamento

A presença de congestionamento significa que a carga é (temporariamente) maior que os recursos (em uma parte da rede) podem manipular. Duas soluções vêm à mente: aumente os recursos ou diminua a carga. Conforme mostrado na Fig. 5-22, essas soluções ções são geralmente aplicadas em diferentes escalas de tempo para evitar o congestionamento ou reaja a ele uma vez que tenha ocorrido.

Consciente do tráfego
roteamento
Rede
provisionamento
Tráfego
estrangulamento
Admissão
ao controle
Carga
derramamento
Mais devagar
(Preventivo)
Mais rápido
(Reativo)

Figura 5-22. Escalas de tempo de abordagens para controle de congestionamento.

A maneira mais básica de evitar o congestionamento é construir uma rede que esteja bem compatível com o tráfego que transporta. Se houver um link de baixa largura de banda no caminho ao longo do qual a maior parte do tráfego é direcionado, é provável que haja congestionamento. Às vezes, recursos

SEC. 5,3

ALGORITMOS DE CONTROLE DE CONGESTÃO

395

pode ser adicionado dinamicamente quando há congestionamento sério, por exemplo, girando em roteadores sobressalentes ou habilitando linhas que normalmente são usadas apenas como backups (para fazer

tolerante a falhas do sistema) ou compra de largura de banda no mercado aberto. Mais frequentemente, links e roteadores que são regularmente muito utilizados são atualizados no início mais oportunidade. Isso é chamado de **provisionamento** e acontece em uma escala de tempo de meses, impulsionado por tendências de tráfego de longo prazo.

Para aproveitar ao máximo a capacidade da rede existente, as rotas podem ser personalizadas para padrões de tráfego que mudam durante o dia conforme os usuários da rede acordam e dormem em diferentes

fusos horários diferentes. Por exemplo, as rotas podem ser alteradas para desviar o tráfego de caminhos muito usados, alterando os pesos dos caminhos mais curtos. Algumas estações de rádio locais

ções têm helicópteros voando em torno de suas cidades para relatar o congestionamento das estradas para

possibilitar que seus ouvintes móveis encaminhem seus pacotes (carros) ao redor pontos de acesso. Isso é chamado **de roteamento ciente do tráfego**. Dividindo o tráfego em vários caminhos também é útil.

No entanto, às vezes não é possível aumentar a capacidade. O único jeito então, vencer o congestionamento é diminuir a carga. Em uma rede de circuito virtual funcionar, novas conexões podem ser recusadas se fizerem com que a rede se torne congestionado. Isso é chamado de **controle de admissão**.

Em uma granularidade mais fina, quando o congestionamento é iminente, a rede pode fornecer feedback para as fontes cujos fluxos de tráfego são responsáveis pelo problema. A rede pode solicitar a essas fontes para restringir seu tráfego, ou pode retardar o próprio tráfego.

Duas dificuldades com esta abordagem são como identificar o início dos congestionamentos e como informar a fonte que precisa desacelerar. Para enfrentar o primeiro problema, os roteadores podem monitorar a carga média, o atraso na fila ou a perda de pacotes. Em tudo

casos, números crescentes indicam um congestionamento crescente.

Para resolver o segundo problema, os roteadores devem participar de um ciclo de feedback com as fontes. Para um esquema funcionar corretamente, a escala de tempo deve ser ajustada com cuidado-

totalmente. Se cada vez que dois pacotes chegam em uma linha, um roteador grita STOP e cada sempre que um roteador ficar ocioso por 20 µs, ele grita GO, o sistema oscilará descontroladamente e

nunca convergem. Por outro lado, se esperar 30 minutos para ter certeza antes dizendo qualquer coisa, o mecanismo de controle de congestionamento vai reagir muito lentamente para ser

de qualquer uso. Fornecer feedback oportuno não é uma questão trivial. Uma preocupação adicional é fazer com que os roteadores enviem mais mensagens quando a rede já está congestionada.

Finalmente, quando tudo mais falha, a rede é forçada a descartar os pacotes que ela não pode entregar. O nome geral para isso é **redução de carga**. Uma boa política para escolher quais pacotes descartar pode ajudar a prevenir o colapso do congestionamento.

5.3.2 Roteamento sensível ao tráfego

A primeira abordagem que examinaremos é o roteamento ciente do tráfego. O roteamento os esquemas que vimos na Seção 5.2 usavam pesos de link fixos. Esses esquemas adaptados às mudanças na topologia, mas não às mudanças na carga. O objetivo de levar carga para

Página 420

396

A CAMADA DE REDE
INDIVÍDUO. 5

conta quando as rotas de computação são para desviar o tráfego de hotspots que serão os primeiros locais na rede a experimentar congestionamento.

A maneira mais direta de fazer isso é definir o peso do link para ser uma função da largura de banda do link (fixa) e atraso de propagação mais a carga medida (variável) ou atraso médio na fila. Os caminhos de menor peso irão, então, favorecer os caminhos que são mais levemente carregado, todo o resto sendo igual.

O roteamento ciente de tráfego foi usado no início da Internet de acordo com este modelo (Khanna e Zinky, 1989). No entanto, existe um perigo. Considere a rede de Fig. 5-23, que é dividido em duas partes, Leste e Oeste, conectadas por dois links, *CF* e *EI*. Suponha que a maior parte do tráfego entre o leste e o oeste esteja usando link *CF* e, como resultado, este link está muito carregado com longos atrasos. Incluindo fila- o atraso no peso usado para o cálculo do caminho mais curto tornará o *EI* mais atraente. Depois que as novas tabelas de roteamento foram instaladas, a maior parte do Leste-Oeste o tráfego passará agora por *EI*, carregando este link. Consequentemente, na próxima atualização, *CF* parecerá ser o caminho mais curto. Como resultado, as tabelas de roteamento podem oscilar tarde descontroladamente, levando a roteamento errático e muitos problemas potenciais.

Oeste
Leste
B

UMA
D
E
C
F
G
H
J
Eu

Figura 5-23. Uma rede na qual as partes Leste e Oeste são conectadas por dois links.

Se a carga for ignorada e apenas a largura de banda e o atraso de propagação forem considerados, este problema não ocorre. Tenta incluir carga, mas alterar pesos dentro uma faixa estreita apenas diminui as oscilações de roteamento. Duas técnicas podem contribuir mas para uma solução de sucesso. O primeiro é o roteamento de caminhos múltiplos, no qual pode haver vários caminhos de uma origem a um destino. Em nosso exemplo, isso significa que o tráfego pode ser distribuído pelos links leste a oeste. O segundo é para o esquema de roteamento para deslocar o tráfego entre as rotas lentamente o suficiente para que seja capaz de convergir, como no esquema de Gallagher (1977).

Dadas essas dificuldades, os protocolos de roteamento da Internet geralmente não ad- apenas suas rotas dependendo da carga. Em vez disso, os ajustes são feitos fora do protocolo de roteamento, alterando lentamente suas entradas. Isso é chamado de **engenharia de tráfego**.

Página 421

SEC. 5,3

ALGORITMOS DE CONTROLE DE CONGESTÃO

397

5.3.3 Controle de Admissão

Uma técnica que é amplamente usada em redes de circuito virtual para manter conges- ção na baía é o **controle de admissão**. A ideia é simples: não configure um novo virtual circuito, a menos que a rede possa transportar o tráfego adicionado sem ficar congestionada ed. Portanto, as tentativas de configurar um circuito virtual podem falhar. Isso é melhor do que o alter-

nativo, pois permitir que mais pessoas entrem quando a rede está ocupada só torna as coisas pior. Por analogia, no sistema telefônico, quando um switch fica sobrecarregado, pratica o controle de admissão não dando tons de discagem.

O truque com esta abordagem é descobrir quando um novo circuito virtual irá levar ao congestionamento. A tarefa é simples na rede telefônica porque da largura de banda fixa das chamadas (64 kbps para áudio não compactado). No entanto, vir- os circuitos virtuais em redes de computadores vêm em todas as formas e tamanhos. Assim, o circuito

deve vir com alguma caracterização de seu tráfego se quisermos aplicar a admissão ao controle.

O tráfego é frequentemente descrito em termos de taxa e forma. O problema de como descrevê-lo de uma forma simples, mas significativa, é difícil porque o tráfego é típico claramente explosiva - a taxa média é apenas metade da história. Por exemplo, tráfego que varia enquanto navegar na web é mais difícil de lidar do que um filme com a mesma taxa de transferência de longo prazo porque as explosões de tráfego da Web são mais provavelmente congestionarão roteadores na rede. Um descritor comumente usado que cap- tura esse efeito é o **balde furado** ou **balde de fichas**. Um balde furado tem dois pa- rametros que limitam a taxa média e o tamanho do burst instantâneo do tráfego.

Visto que baldes furados são amplamente usados para qualidade de serviço, iremos examiná-los em detalhes na Seç. 5,4

Munida de descrições de tráfego, a rede pode decidir se aceita o novo circuito virtual. Uma possibilidade é a rede reservar capacidade suficiente ao longo dos caminhos de cada um de seus circuitos virtuais esse congestionamento não ocorrerá. Nisso

caso, a descrição do tráfego é um acordo de serviço para o que a rede irá garantir

ante seus usuários. Evitamos o congestionamento, mas mudamos para o tópico relacionado de qualidade do serviço um pouco cedo; voltaremos a ele na próxima seção.

Mesmo sem oferecer garantias, a rede pode usar descrições de tráfego para Controle de admissão. A tarefa é estimar quantos circuitos cabem dentro a capacidade de transporte da rede sem congestionamento. Suponha que circuitos que podem gerar tráfego a taxas de até 10 Mbps, todos passam pelos mesmos 100-Link físico de Mbps. Quantos circuitos devem ser admitidos? Claramente, 10 circuitos pode ser admitido sem risco de congestionamento, mas isso é um desperdício no caso normal uma vez que raramente pode acontecer que todos os 10 estejam transmitindo em alta velocidade ao mesmo tempo.

Em redes reais, medições de comportamento passado que capturam as estatísticas de as transmissões podem ser usadas para estimar o número de circuitos a admitir, para negociar melhor desempenho para risco aceitável.

O controle de admissão também pode ser combinado com o roteamento ciente do tráfego, considerando

ering rotas em torno de pontos de acesso de tráfego como parte do procedimento de configuração. Por exemplo,

Página 422

398

A CAMADA DE REDE

INDIVÍDUO. 5

considere a rede ilustrada na Figura 5.24 (a), na qual dois roteadores estão congestionados.

ed, conforme indicado.

UMA
Congestionamento
Virtual
o circuito
Congestionamento
B
UMA
B
(uma)
(b)

Figura 5-24. (a) Uma rede congestionada. (b) A parte da rede que não é congestionado. Um circuito virtual de *A* a *B* também é mostrado.

Suponha que um host conectado ao roteador *A* deseja configurar uma conexão com um host ligado ao router *B*. Normalmente, esta conexão passaria por um dos roteadores congestionados. Para evitar essa situação, podemos redesenhar a rede como mostrado em Figura 5-24 (b), omitindo os roteadores congestionados e todas as suas linhas. A linha tracejada mostra uma possível rota para o circuito virtual que evita os roteadores congestionados.

Shaikh et al. (1999) fornecem um projeto para este tipo de roteamento sensível à carga.

5.3.4 Estrangulamento de tráfego

Na Internet e em muitas outras redes de computadores, os remetentes ajustam seus transmissões para enviar tanto tráfego quanto a rede puder entregar prontamente. Neste cenário, a rede visa operar um pouco antes do início do congestionamento. Quando congestionamento é iminente, ele deve dizer aos remetentes para reduzir suas transmissões e diminuir baixa. Esse feedback é normal, e não uma situação excepcional. o termo **prevenção de congestionamento** às vezes é usado para contrastar este ponto operacional com aquele em que a rede ficou (excessivamente) congestionada.

Vejamos agora algumas abordagens para limitar o tráfego que podem ser usadas em redes de datagramas e redes de circuitos virtuais. Cada abordagem deve resolver dois problemas. Primeiro, os roteadores devem determinar quando o congestionamento está se aproximando,

idealmente antes de chegar. Para fazer isso, cada roteador pode monitorar continuamente o recursos que está usando. Três possibilidades são a utilização dos links de saída, o buffering de pacotes enfileirados dentro do roteador, e o número de pacotes que são perdido devido a buffer insuficiente. Dessa possibilidades, a segunda é a muito útil. As médias de utilização não são responsáveis diretamente pela explosão de

SEC. 5,3

ALGORITMOS DE CONTROLE DE CONGESTÃO

399

a maioria do tráfego - uma utilização de 50% pode ser baixa para tráfego regular e muito alta para tráfego altamente variável. A contagem de perdas de pacotes chega tarde demais. O congestionamento al-

pronto definido no momento em que os pacotes são perdidos.

O atraso de enfileiramento dentro dos roteadores captura diretamente qualquer experiência de congestionamento

encedido por pacotes. Deve ser baixo na maioria das vezes, mas vai pular quando houver um explosão de tráfego que gera uma lista não processada. Para manter uma boa estimativa do atraso de fila, d , uma amostra do comprimento instantâneo da fila, s , pode ser feita por iodicamente ed atualizado de acordo com

$$d_{\text{novo}} = \alpha d_{\text{antigo}} + (1 - \alpha) s$$

onde a constante α determina a rapidez com que o roteador esquece o histórico recente. Isto é chamado **EWMA** (**E**xponentially **W**eighted **M**oving **A**verage). Isso suaviza flutuações e é equivalente a um filtro passa-baixa. Sempre que d se move acima do limite, o roteador observa o início do congestionamento.

O segundo problema é que os roteadores devem fornecer feedback oportuno para o envio ers que estão causando o congestionamento. Há congestionamento na rede, mas aliviar o congestionamento requer ação em nome dos remetentes que estão usando a rede trabalhos. Para fornecer feedback, o roteador deve identificar os remetentes apropriados. isto deve então avisá-los com cuidado, sem enviar muitos pacotes a mais para o rede congestionada pronta. Diferentes esquemas usam diferentes mecanismos de feedback, como iremos descrever agora.

Choke Packets

A maneira mais direta de notificar um remetente sobre o congestionamento é informá-lo diretamente. No

esta abordagem, o roteador seleciona um pacote congestionado e envia um **pacote bloqueador** de volta ao host de origem, fornecendo a ele o destino encontrado no pacote. O original o pacote pode ser etiquetado (um bit de cabeçalho está ligado) para que não gere nenhum mais pacotes de bloqueio mais adiante ao longo do caminho e então encaminhados da maneira usual. Para evitar o aumento da carga na rede durante um período de congestionamento, o roteador só pode enviar pacotes de bloqueio a uma taxa baixa.

Quando o host de origem obtém o pacote choke, é necessário reduzir o tráfego enviado para o destino especificado, por exemplo, em 50%. Em uma rede de datagramas, simplesmente pegar os pacotes aleatoriamente quando há congestionamento pode causar pacotes de bloqueio a serem enviados para remetentes rápidos, porque eles terão a maioria dos pacotes

na fila. O feedback implícito neste protocolo pode ajudar a prevenir o congestionamento ainda assim, não estrangular nenhum remetente, a menos que cause problemas. Pelo mesmo motivo, é como

Provavelmente, vários pacotes de bloqueio serão enviados para um determinado host e destino. o o host deve ignorar esses chokes adicionais pelo intervalo de tempo fixo até que seu redução no tráfego entra em vigor. Após esse período, outros pacotes de choke indicam que a rede ainda está congestionada.

Um exemplo de pacote de choke usado no início da Internet é o SOURCE-Mensagem QUENCH (Postel, 1981). Nunca pegou, em parte porque o

400

A CAMADA DE REDE

INDIVÍDUO. 5

circunstâncias em que foi gerado e o efeito que teve não foram claramente Especificadas. A Internet moderna usa um design de notificação alternativo que iremos descreva a seguir.

Notificação explícita de congestionamento

Em vez de gerar pacotes adicionais para alertar sobre congestionamento, um roteador pode etiquetar qualquer pacote que ele encaminhe (definindo um bit no cabeçalho do pacote) para sinalizar que ele

está enfrentando congestionamento. Quando a rede entrega o pacote, o destino pode notar que há congestionamento e informar o remetente quando ele enviar um pacote de resposta-

et. O remetente pode então controlar suas transmissões como antes.

Este projeto é chamado de **ECN (Notificação Explícita de Congestionamento)** e é usado em a Internet (Ramakrishnan et al., 2001). É um refinamento do congestionamento inicial protocolos de sinalização, principalmente o esquema de feedback binário de Ramakrishnan e Jain (1988) que foi usado na arquitetura DECNET. Dois bits no pacote IP cabeçalho são usados para registrar se o pacote experimentou congestionamento. Pacotes não estão marcados quando são enviados, conforme ilustrado na Figura 5.25. Se algum dos roteadores

eles passam congestionados, esse roteador irá então marcar o pacote como tendo experimentou congestionamento ao ser encaminhado. O destino irá então ecoar qualquer marca de volta ao remetente como um sinal de congestionamento explícito em seu próximo pacote de resposta.

Isso é mostrado com uma linha tracejada na figura para indicar que acontece acima do Nível de IP (por exemplo, em TCP). O remetente deve então acelerar suas transmissões, como no caso de pacotes de choke.

Sinal de congestionamento
Hospedeiro
Marcado
pacote
Hospedeiro
Pacote
Congestionado
roteador

Figura 5-25. Notificação explícita de congestionamento

Contrapressão hop-by-hop

Em altas velocidades ou em longas distâncias, muitos novos pacotes podem ser transmitidos após o congestionamento ter sido sinalizado por causa do atraso antes de o sinal entrar em vigor efeito. Considere, por exemplo, um host em São Francisco (roteador *A* na Figura 5.26) que é enviar tráfego para um host em Nova York (roteador *D* na Figura 5-26) na velocidade OC-3 de 155 Mbps. Se o host de Nova York começar a ficar sem buffers, demorará cerca de 40 milissegundos para um pacote de estrangulamento voltar a São Francisco para dizer para diminuir a velocidade. A

A indicação ECN demorará ainda mais porque é entregue via destino.

A propagação do pacote de choke é ilustrada como a segunda, terceira e quarta etapas em

SEC. 5,3

ALGORITMOS DE CONTROLE DE CONGESTÃO

401

Fig. 5-26 (a). Nesses 40 ms, outros 6,2 megabits terão sido enviados. Mesmo se o host em São Francisco desliga completamente imediatamente, os 6,2 megabits em o tubo continuarão a derramar e terá que ser tratado. Só no sétimo no diagrama da Figura 5-26 (a), o roteador de Nova York notará um fluxo mais lento. Uma abordagem alternativa é fazer com que o pacote de choke tenha efeito a cada salto que passa, conforme mostrado na seqüência da Figura 5.26 (b). Aqui, assim que o estrangulamento pacote atinge *F*, *F* é necessária para reduzir o fluxo para *D*. Fazer isso vai voltar quire *F* para dedicar mais buffers à conexão, uma vez que a fonte ainda está enviando a todo vapor, mas dá alívio imediato a *D*, como um remédio para dor de cabeça em um comercial de televisão. Na próxima etapa, o pacote de choke chega a *E*, que diz a *E* para reduzir o fluxo para *F*. Esta ação coloca uma maior demanda nos buffers de *E*, mas dá a *F* alívio imediato. Finalmente, o pacote de estrangulamento atinge *A* e o fluxo genuinamente diminui.

O efeito líquido deste esquema hop-by-hop é fornecer alívio rápido no ponto

de congestionamento, ao preço de usar mais buffers upstream. Desta forma, a gestão pode ser cortado pela raiz sem perder nenhum pacote. A ideia é discutido em detalhes por Mishra et al. (1996).

5.3.5 Redução de carga

Quando nenhum dos métodos acima faz o congestionamento desaparecer, os roteadores podem tirar a artilharia pesada: derramamento de carga. A **redução de carga** é uma maneira elegante de dizendo que quando os roteadores estão sendo inundados por pacotes que eles não podem lidar, eles simplesmente os jogam fora. O termo vem do mundo da energia elétrica geração, onde se refere à prática das concessionárias apagarem intencionalmente certas áreas para evitar que toda a grade entre em colapso em dias quentes de verão, quando a demanda por eletricidade excede em muito a oferta.

A questão chave para um roteador se afogando em pacotes é quais pacotes devem ser descartados. A escolha preferida pode depender do tipo de aplicativo que usa a rede.

Para uma transferência de arquivo, um pacote antigo vale mais do que um novo. Isto é porque descartar o pacote 6 e manter os pacotes de 7 a 10, por exemplo, só forçará o receptor faça mais trabalho para armazenar dados que ele ainda não pode usar. Em contraste, para mídia em tempo real, um novo pacote vale mais do que um antigo. Isto é porque os pacotes tornam-se inúteis se atrasam e perdem o momento em que devem ser reproduzido para o usuário.

A primeira política (a velha é melhor do que a nova) costuma ser chamada de **vinho** e a última (novo é melhor do que velho) é frequentemente chamado de **leite** porque a maioria das pessoas prefere beber leite novo e vinho velho do que a alternativa.

Uma redução de carga mais inteligente requer a cooperação dos remetentes. Um exemplo são os pacotes que transportam informações de roteamento. Esses pacotes são mais importantes

do que pacotes de dados regulares porque eles estabelecem rotas; se eles forem perdidos, a rede o trabalho pode perder conectividade. Outro exemplo é que algoritmos de compressão vídeo, como MPEG, transmite periodicamente um quadro inteiro e, em seguida, envia

Página 426

402

A CAMADA DE REDE INDIVÍDUO. 5

(uma)

(b)

Choke

Choke

B

C

UMA

D

E

F

Choke

Reducido

fluxo

O fluxo está parado
na taxa máxima

Fluxo é
reduzido

B

C

UMA

D

E

F

Fluxo pesado

Choke

Choke

Choke

Reducido

fluxo

Figura 5-26. (a) Um pacote de choke que afeta apenas a fonte. (b) Um pacote de estrangulamento
que afeta cada salto que passa.

SEC. 5,3

ALGORITMOS DE CONTROLE DE CONGESTÃO

403

quadros como diferenças do último quadro completo. Neste caso, descartando um pacote que faz parte de uma diferença é preferível descartar um que seja parte de um quadro completo antes de porque os pacotes futuros dependem do quadro completo.

Para implementar uma política de descarte inteligente, os aplicativos devem marcar sua embalagem eis para indicar à rede o quanto importante eles são. Então, quando os pacotes precisam ser descartado, os roteadores podem primeiro descartar os pacotes da classe menos importante, depois a

próxima aula mais importante e assim por diante.

Claro, a menos que haja algum incentivo significativo para evitar marcar cada pacote como MUITO IMPORTANTE - NUNCA, NUNCA DESCARTE, ninguém vai fazer isso.

Freqüentemente, a contabilidade e o dinheiro são usados para desencorajar a marcação frívola. Para ex-

amplo, a rede pode permitir que os remetentes enviem mais rápido do que o serviço que compraram permite se eles marcarem os pacotes em excesso como de baixa prioridade. Tal estratégia não é uma má ideia porque torna mais eficiente o uso de recursos ociosos, permitindo que os hosts usá-los enquanto ninguém mais estiver interessado, mas sem estabelecer o direito de eles quando os tempos ficam difíceis.

Detectão Inicial Aleatória

Lidar com o congestionamento quando ele começa é mais eficaz do que deixá-lo atrapalhar o trabalho e tentar lidar com isso. Esta observação leva a um interessante torção na redução de carga, que é descartar pacotes antes de todo o buffer o espaço está realmente esgotado.

A motivação para essa ideia é que a maioria dos hosts da Internet ainda não obtém congestionamentos de conexão de roteadores na forma de ECN. Em vez disso, a única indicação confiável de congestionamento que os hosts obtêm da rede é a perda de pacotes. Afinal, é difícil culto para construir um roteador que não descarta pacotes quando está sobrecarregado. Transporte protocolos como o TCP são, portanto, conectados para reagir à perda como congestionamento, desacelerando

baise a fonte em resposta. O raciocínio por trás dessa lógica é que o TCP foi desenhado para redes com fio e redes com fio são muito confiáveis, portanto, pacotes perdidos são principalmente devido a saturações de buffer em vez de erros de transmissão. Links sem fio deve recuperar erros de transmissão na camada de enlace (para que não sejam vistos na rede camada de trabalho) para funcionar bem com TCP.

Esta situação pode ser explorada para ajudar a reduzir o congestionamento. Por ter roteadores descartar pacotes mais cedo, antes que a situação se torne desesperadora, há tempo para a fonte para agir antes que seja tarde demais. Um algoritmo popular para fazer isso é chamado **RED (Random Early Detection)** (Floyd e Jacobson, 1993). Para determinar quando começar a descartar, os roteadores mantêm uma média de execução de sua fila comprimentos. Quando o comprimento médio da fila em algum link excede um limite, o link é considerado congestionado e uma pequena fração dos pacotes são descartados no random. A seleção de pacotes aleatoriamente torna mais provável que os remetentes mais rápidos ver um pacote cair; esta é a melhor opção, pois o roteador não pode dizer qual fonte está causando a maioria dos problemas em uma rede de datagramas. O remetente afetado irá observe a perda quando não há reconhecimento e, em seguida, o protocolo de transporte

404

A CAMADA DE REDE

INDIVÍDUO. 5

vai desacelerar. O pacote perdido está, portanto, entregando a mesma mensagem que um estrangulamento

pacote, mas implicitamente, sem o roteador enviar qualquer sinal explícito.

Os roteadores RED melhoram o desempenho em comparação com os roteadores que descartam apenas pacotes quando seus buffers estão cheios, embora possam precisar de ajuste para funcionar bem. Para exemplo, o número ideal de pacotes a serem descartados depende de quantos remetentes precisam ser notificado de congestionamento. No entanto, ECN é a opção preferida, se estiver disponível. Funciona exatamente da mesma maneira, mas fornece um sinal de congestionamento explicitamente em vez de uma perda; RED é usado quando os hosts não podem receber sinais explícitos.

5.4 QUALIDADE DE SERVICO

As técnicas que vimos nas seções anteriores são projetadas para reduzir congestionamento e melhorar o desempenho da rede. No entanto, existem aplicativos (e clientes) que exigem garantias de desempenho mais fortes da rede do que "o melhor que poderia ser feito nas circunstâncias." em particular, muitas vezes precisam de uma taxa de transferência mínima e latência máxima para trabalhos. Nesta seção, continuaremos nosso estudo de desempenho de rede, mas agora com um foco mais nítido em maneiras de fornecer qualidade de serviço que corresponda a necessidades do aplicativo. Esta é uma área em que a Internet está passando por um longo prazo melhoria.

Uma solução fácil para fornecer um serviço de boa qualidade é construir uma rede com capacidade suficiente para qualquer tráfego que for lançado nele. O nome desta solução é **superprovisionamento**. A rede resultante transportará o tráfego de aplicativos sem perdas significativas e, assumindo um esquema de roteamento decente, entregará o pacote ets com baixa latência. O desempenho não existe nada melhor do que isso. Para alguns extensão, o sistema telefônico é superprovisionado porque é raro pegar um telefone telefone e não receba um tom de discagem instantaneamente. Simplesmente há tanta capacidade disponível

capaz de quase sempre essa demanda pode ser atendida.

O problema dessa solução é que ela é cara. É basicamente resolver um problema jogando dinheiro nisso. Mecanismos de qualidade de serviço permitem que uma rede com menos capacidade, atenda aos requisitos do aplicativo da mesma forma a um custo menor. Além disso, o superprovisionamento é baseado no tráfego esperado. Todas as apostas estão canceladas se o

o padrão de tráfego muda muito. Com mecanismos de qualidade de serviço, a rede trabalho pode honrar as garantias de desempenho que faz mesmo quando o tráfego picos, ao custo de recusar alguns pedidos.

Quatro questões devem ser abordadas para garantir a qualidade do serviço:

1. O que os aplicativos precisam da rede.
2. Como regular o tráfego que entra na rede.
3. Como reservar recursos em roteadores para garantir o desempenho.
4. Se a rede pode aceitar mais tráfego com segurança.

Página 429

SEC. 5,4

QUALIDADE DE SERVIÇO

405

Nenhuma técnica lida com eficiência com todos esses problemas. Em vez disso, uma variedade de técnicas foram desenvolvidas para uso na camada de rede (e transporte).

Soluções práticas de qualidade de serviço combinam várias técnicas. Para este fim, iremos descrever duas versões de qualidade de serviço para a Internet chamadas Serviços Integrados e Diferenciados.

5.4.1 Requisitos do Aplicativo

Um fluxo de pacotes de uma fonte para um destino é chamado de **fluxo** (Clark, 1988). Um fluxo pode ser todos os pacotes de uma conexão em uma conexão orientada rede, ou todos os pacotes enviados de um processo para outro processo em uma rede sem conexão. As necessidades de cada fluxo podem ser caracterizadas por quatro primários mary: largura de banda, atraso, jitter e perda. Juntos, eles determinam o **QoS (Quality of Service)** que o fluxo exige.

Vários aplicativos comuns e o rigor de sua rede re-

Os procedimentos estão listados na Figura 5-27. Observe que os requisitos de rede são menos difíceis exigindo que os requisitos do aplicativo nos casos em que o aplicativo pode exigir comprovar no serviço prestado pela rede. Em particular, as redes não precisam para ser sem perdas para transferência confiável de arquivos, e eles não precisam entregar pacotes com atrasos idênticos para reprodução de áudio e vídeo. Alguma quantidade de perda pode ser reparada com retransmissões, e alguma quantidade de jitter pode ser suavizada pelo buffer pacotes no receptor. No entanto, não há nada que os aplicativos possam fazer para remediar a situação se a rede fornecer pouca largura de banda ou muito atraso.

Inscrição

Largura de banda

Demora

Jitter

Perda

O email

Baixo

Baixo

Baixo

Médio

Compartilhamento de arquivos

Alto

Baixo

Baixo

Médio

Acesso à web

Médio

Médio

Baixo

Médio

Login remoto

Baixo

Médio

Médio

Médio

Áudio sob demanda

Baixo

Baixo

Alto

Baixo

Vídeo sob demanda

Alto

Baixo

Alto

Baixo

Telefonia

Baixo

Alto

Alto

Baixo

Vídeo conferência

Alto

Alto

Alto

Baixo

Figura 5-27. Rigor dos requisitos de qualidade de serviço dos aplicativos.

Os aplicativos diferem em suas necessidades de largura de banda, com e-mail, áudio em todos formulários e login remoto não precisam de muito, mas compartilhamento de arquivos e vídeo em todas as formas precisando de muito.

Mais interessantes são os requisitos de atraso. Aplicativos de transferência de arquivos, incluindo e-mail e vídeo, não são sensíveis a atrasos. Se todos os pacotes estão atrasados unicamente por alguns segundos, nenhum dano é causado. Aplicativos interativos, como Web

surf e login remoto são mais sensíveis a atrasos. Aplicativos em tempo real, como como telefonia e videoconferência, têm requisitos de atraso estritos. Se todo o palavras em uma chamada telefônica são atrasadas por muito tempo, os usuários encontrarão a necção inaceitável. Por outro lado, reproduzir arquivos de áudio ou vídeo de um servidor ver não requer baixo atraso.

A variação (ou seja, desvio padrão) no atraso ou tempos de chegada do pacote é chamado **jitter**. As três primeiras aplicações na Fig. 5-27 não são sensíveis à embalagem ets chegando com intervalos de tempo irregulares entre eles. Login remoto é algum- o que é sensível a isso, já que as atualizações na tela aparecerão em pequenas rajadas se o conexão sofre muito jitter. O vídeo e especialmente o áudio são extremamente sensíveis tive a jitter. Se um usuário está assistindo a um vídeo pela rede e os frames são todos atrasado por exatamente 2.000 segundos, nenhum dano é causado. Mas se o tempo de transmissão varia aleatoriamente entre 1 e 2 segundos, o resultado será terrível, a menos que o ap- alicatura esconde o jitter. Para áudio, um jitter de até mesmo alguns milissegundos é claramente audível.

Os primeiros quatro aplicativos têm requisitos mais rigorosos sobre perda do que aud- io e vídeo porque todos os bits devem ser entregues corretamente. Este objetivo é geralmente um- gerada com retransmissões de pacotes que são perdidos na rede pela transmissão camada de esporte. Este é um trabalho perdido; seria melhor se a rede recusasse pacotes era provável que perdesse em primeiro lugar. Aplicativos de áudio e vídeo podem tolerar alguns pacotes perdidos sem retransmissão porque as pessoas não notam curtos pausas ou quadros pulados ocasionais.

Para acomodar uma variedade de aplicações, as redes podem suportar diferentes categorias de QoS. Um exemplo influente vem de redes ATM, que já fizeram parte de uma grande visão de networking, mas desde então se tornaram um nicho tecnologia. Eles apoiam:

1. Taxa de bits constante (por exemplo, telefonia).
2. Taxa de bits variável em tempo real (por exemplo, videoconferência compactada).
3. Taxa de bits variável não em tempo real (por exemplo, assistir a um filme sob demanda).
4. Taxa de bits disponível (por exemplo, transferência de arquivo).

Essas categorias também são úteis para outros fins e outras redes. Constante taxa de bits é uma tentativa de simular um fio, fornecendo uma largura de banda uniforme e um atraso uniforme. A taxa de bits variável ocorre quando o vídeo é compactado, com alguns quadros comprimindo mais do que outros. Enviar uma moldura com muitos detalhes pode exigir o envio de muitos bits, enquanto um tiro de uma parede branca pode comprimir ex- tremadamente bem. Os filmes sob demanda não são realmente em tempo real porque alguns segundos

de vídeo pode ser facilmente armazenado em buffer no receptor antes do início da reprodução, então jitter

a rede apenas faz com que a quantidade de vídeo armazenado, mas não reproduzido, varie.

A taxa de bits disponível é para aplicativos como e-mail que não são sensíveis a atrasos ou jitter e ocuparão toda a largura de banda que conseguirem.

5.4.2 Traffic Shaping

Antes que a rede possa oferecer garantias de QoS, ela deve saber o que é o tráfego sendo garantido. Na rede telefônica, essa caracterização é simples. Para exemplo, uma chamada de voz (em formato não compactado) precisa de 64 kbps e consiste em um Amostra de 8 bits a cada 125 µseg. No entanto, o tráfego nas redes de dados é **intermitente**. É típico

chega a taxas não uniformes conforme a taxa de tráfego varia (por exemplo, videoconferência

com compressão), os usuários interagem com os aplicativos (por exemplo, navegando em uma nova Web página) e os computadores alternam entre as tarefas. Picos de tráfego são mais difíceis de lidar com o tráfego de taxa constante porque podem preencher buffers e fazer com que os pacotes estar perdido.

A modelagem de tráfego é uma técnica para regular a taxa média e rajadas de um fluxo de dados que entra na rede. O objetivo é permitir que os aplicativos transmitem uma grande variedade de tráfego que atende às suas necessidades, incluindo alguns bursts, ainda

têm uma maneira simples e útil de descrever os possíveis padrões de tráfego para a rede trabalhos. Quando um fluxo é configurado, o usuário e a rede (ou seja, o cliente e o provedor) concordar com um determinado padrão de tráfego (ou seja, formato) para esse fluxo. Na verdade, o

cliente diz ao provedor " Meu padrão de transmissão será semelhante a este; pode você lida com isso? "

Às vezes, este contrato é chamado de **SLA (Service Level Agreement)**, especialmente quando é feito sobre fluxos agregados e longos períodos de tempo, como todo o tráfego para um determinado cliente. Contanto que o cliente cumpra sua parte de a barganha e só envia pacotes de acordo com o contrato acordado, o provedor promete entregá-los todos em tempo hábil.

A modelagem de tráfego reduz o congestionamento e, portanto, ajuda a rede a cumprir sua promessa. Porém, para fazer funcionar, há também a questão de como o provedor pode dizer se o cliente está seguindo o acordo e o que fazer se o cliente não é. Pacotes que excedam o padrão acordado podem ser descartados pela rede, ou eles podem ser marcados como tendo prioridade mais baixa. O monitoramento de um fluxo de tráfego é chamado

policimento de tráfego .

A modelagem e o policimento não são tão importantes para transferências ponto a ponto e outras transferências que vai consumir toda e qualquer largura de banda disponível, mas eles são de grande importância para dados em tempo real, como conexões de áudio e vídeo, que têm requisitos de qualidade de serviço rigorosos.

Leaky and Token Buckets

Já vimos uma maneira de limitar a quantidade de dados de um aplicativo envia: a janela deslizante, que usa um parâmetro para limitar a quantidade de dados trânsito em um determinado momento, o que indiretamente limita a taxa. Agora vamos olhar para um maneira mais geral de caracterizar o tráfego, com o balde furado e o balde de tokens algoritmos. As formulações são ligeiramente diferentes, mas fornecem um resultado equivalente.

Tente imaginar um balde com um pequeno orifício no fundo, conforme ilustrado em Fig. 5-28 (b). Não importa a taxa em que a água entra no balde, o fluxo de saída está em uma taxa constante, R , quando há água no balde e zero quando o balde está vazia. Além disso, uma vez que o balde está cheio até a capacidade B , qualquer água adicional entra

derramar pelos lados e se perder.

Verifica
balde
aqui
Hospedeiro
Pacotes
Taxa
R
B
B
Taxa
R

Tirar
 água / tokens
 Colocar em
 água
 Rede
 (uma)
 (b)
 (c)

Figura 5-28. (a) Moldar pacotes. (b) Um balde furado. (c) Um token bucket.

Este balde pode ser usado para moldar ou policiar os pacotes que entram na rede, como mostrado na Fig. 5-28 (a). Conceitualmente, cada host está conectado à rede por um interface contendo um balde furado. Para enviar um pacote para a rede, deve ser possível colocar mais água no balde. Se um pacote chega quando o balde está cheio, o pacote deve ser colocado na fila até que vaze água suficiente para segurá-lo ou ser descartado. O primeiro pode acontecer em um host moldando seu tráfego para a rede como parte do sistema operacional. O último pode acontecer em hardware em um provedor interface de rede que está monitorando o tráfego que entra na rede. Esta técnica foi proposto por Turner (1986) e é chamado de **algoritmo do balde furado**.

Uma formulação diferente, mas equivalente, é imaginar a interface de rede como um balde que está sendo enchido, conforme mostrado na Fig. 5-28 (c). A torneira está funcionando a uma taxa R

e o balde tem capacidade B , como antes. Agora, para enviar um pacote, devemos ser capaz de tirar água, ou tokens, como o conteúdo é comumente chamado, fora do balde (em vez de colocar água no balde). Não mais do que um número fixo de tokens, B , podem se acumular no balde, e se o balde estiver vazio, devemos espere até que mais tokens cheguem antes de enviarmos outro pacote. Este algoritmo é chamado de **algoritmo de token bucket**.

Leaky e token buckets limitam a taxa de longo prazo de um fluxo, mas permitem curto termo estoura até um comprimento máximo regulado para passar inalterado e sem sofrer atrasos artificiais. Grandes rajadas serão suavizadas por um vazamento balde modelador de tráfego para reduzir o congestionamento na rede. Por exemplo, imagine que um computador pode produzir dados a até 1000 Mbps (125 milhões de bytes / s) e que o primeiro link da rede também funciona nessa velocidade. O padrão de tráfego o host gera é mostrado na Figura 5.29 (a). Este padrão é intermitente. A média

Página 433

SEC. 5,4 QUALIDADE DE SERVIÇO **409**

a taxa de mais de um segundo é de 200 Mbps, embora o host envie um burst de 16.000 KB na velocidade máxima de 1000 Mbps (para 1/8 do segundo).

25 MB / s para
 250 mseg
 125 MB / s para
 125 mseg
 Tempo (mseg)
 16.000
 1000
 Taxa (Mbps)
 (uma)
 (d)
 (b)
 (e)
 (c)
 (f)
 1000
 Bucket (KB)
 Com $R = 25$ MB / s, $B = 0$
 Com $R = 25$ MB / s,
 $B = 9600$ KB
 Balde sempre vazio
 Baldes vazios,
 tráfego atrasado
 Tempo (mseg)
 1000
 9600
 0

Figura 5-29. (a) Tráfego de um host. Saída moldada por um balde simbólico de taxa 200 Mbps e capacidade (b) 9600 KB e (c) 0 KB. Nível de balde de token para shap-

com taxa de 200 Mbps e capacidade (d) 16.000 KB, (e) 9600 KB e (f) 0 KB.

Agora, suponha que os roteadores possam aceitar dados na velocidade máxima apenas por um curto período

intervalos, até que seus buffers se enchem. O tamanho do buffer é 9600 KB, menor que o estouro de tráfego. Para intervalos longos, os roteadores funcionam melhor em taxas não superiores a 200

Mbps (digamos, porque essa é toda a largura de banda fornecida ao cliente). As implicações são que se o tráfego for enviado neste padrão, parte dele será descartado na rede

funciona porque não cabe nos buffers dos roteadores.

Para evitar essa perda de pacotes, podemos moldar o tráfego no host com um token balde. Se usarmos uma taxa, R , de 200 Mbps e uma capacidade, B , de 9600 KB, o tráfego cairá dentro do que a rede pode suportar. A saída deste token bucket é mostrado na Figura 5-29 (b). O host pode enviar aceleração total a 1000 Mbps por um curto enquanto até que tenha esvaziado o balde. Então ele tem que cortar para 200 Mbps até o explosão foi enviada. O efeito é espalhar a explosão ao longo do tempo porque foi muito grande para manusear tudo de uma vez. O nível do token bucket é mostrado na Fig. 5-29 (e). Ele começa cheio e é esgotado pelo burst inicial. Quando chega a zero, novos pacotes podem ser enviados apenas na taxa em que o buffer está enchendo; pode haver não há mais rajadas até que o balde se recupere. O balde se enche quando não há tráfego sendo enviado e permanece estável quando o tráfego está sendo enviado na taxa de preenchimento. Também podemos moldar o tráfego para ter menos rajadas. Fig. 5-29 (c) mostra a saída de um token bucket com $R = 200$ Mbps e uma capacidade de 0. Este é o caso extremo

Página 434

410

A CAMADA DE REDE

INDIVÍDUO. 5

em que o tráfego foi completamente suavizado. Não são permitidos rajadas, e o tráfego entra na rede em uma taxa constante. O nível de balde correspondente, mostrado na Figura 5.29 (f), está sempre vazio. O tráfego está sendo enfileirado no host para liberação na rede e sempre há um pacote esperando para ser enviado quando é permitido-ed.

Finalmente, a Fig. 5-29 (d) mostra o nível do balde para um token balde com $R = 200$ Mbps e capacidade de $B = 16.000$ KB. Este é o menor token bucket através pelo qual o tráfego passa inalterado. Ele pode ser usado em um roteador na rede para polícia o tráfego que o host envia. Se o host está enviando tráfego em conformidade com o token bucket no qual concordou com a rede, o tráfego caberá

por meio desse mesmo token bucket executado no roteador na extremidade da rede. Se o host envia em uma taxa mais rápida ou mais rápida, o balde de token ficará sem água. Se este acontecer, um vigilante de tráfego saberá que o tráfego não é conforme descrito. Então será descartar os pacotes em excesso ou diminuir sua prioridade, dependendo do design do a rede. Em nosso exemplo, o balde esvazia apenas momentaneamente, no final de

a rajada inicial, então recupera o suficiente para a próxima rajada.

Leaky e token buckets são fáceis de implementar. Vamos agora descrever o operação de um token bucket. Mesmo que tenhamos descrito água fluindo continuamente dentro e fora do balde, as implementações reais devem funcionar com quantidades. Um token bucket é implementado com um contador para o nível do balde. O contador é avançado em unidades $R / \Delta T$ a cada tique do relógio de ΔT segundos. Isso seria de 200 Kbit a cada 1 ms em nosso exemplo acima. Cada vez que uma unidade de o tráfego é enviado para a rede, o contador é diminuído e o tráfego pode ser enviado até que o contador chegue a zero.

Quando os pacotes são todos do mesmo tamanho, o nível do balde pode apenas ser contado em pacotes (por exemplo, 200 Mbit são 20 pacotes de 1250 bytes). No entanto, frequentemente variável pacotes de tamanhos diferentes estão sendo usados. Nesse caso, o nível do balde é contado em bytes. E se

a contagem de bytes residual é muito baixa para enviar um pacote grande, o pacote deve esperar até a próxima marca (ou ainda mais, se a taxa de preenchimento for pequena).

Calcular o comprimento da explosão máxima (até que o balde esvazie) é um pouco complicado. É mais longo do que apenas 9600 KB dividido por 125 MB / s porque enquanto o burst está sendo gerado, mais tokens chegam. Se chamarmos o comprimento do burst de S

s., a taxa de saída máxima M bytes / s, a capacidade do token bucket B bytes e a taxa de chegada de token R bytes / s, podemos ver que uma explosão de saída contém um máximo mae de bytes $B + RS$. Também sabemos que o número de bytes em um máximo rajada de velocidade de duração S segundos é MS . Portanto, temos

$$B + RS = MS$$

Podemos resolver esta equação para obter $S = B / (M - R)$. Para nossos parâmetros de $B = 9600$ KB, $M = 125$ MB / seg e $R = 25$ MB / seg, obtemos um tempo de burst de cerca de 94 mseg.

Um problema potencial com o algoritmo de token bucket é que ele reduz grandes rajadas baixo para a taxa de longo prazo R . Freqüentemente, é desejável reduzir o pico taxa, mas sem descer para a taxa de longo prazo (e também sem aumentar a

Página 435

SEC. 5,4

QUALIDADE DE SERVIÇO

411

taxa de longo prazo para permitir mais tráfego na rede). Uma maneira de ficar mais suave o tráfego é inserir um segundo token bucket após o primeiro. A taxa do segundo balde deve ser muito mais alto do que o primeiro. Basicamente, o primeiro balde caracteriza o tráfego, fixando sua taxa média, mas permitindo alguns bursts. O segundo o balde reduz a taxa de pico na qual os bursts são enviados para a rede. Para exemplo, se a taxa do segundo token bucket for definida como 500 Mbps e a capacidade ty é definido como 0, o burst inicial entrará na rede a uma taxa de pico de 500 Mbps, que é inferior à taxa de 1000 Mbps que tínhamos anteriormente.

Usar todos esses baldes pode ser um pouco complicado. Quando token buckets são usados para modelagem de tráfego nos hosts, os pacotes são enfileirados e atrasados até que os buckets permitam para serem enviados. Quando token buckets são usados para policiamento de tráfego em roteadores no

rede, o algoritmo é simulado para garantir que nenhum outro pacote seja enviado do que o permitido. No entanto, essas ferramentas fornecem maneiras de moldar o tráfego da rede fic em formas mais gerenciáveis para auxiliar no atendimento de qualidade de serviço quirements.

5.4.3 Programação de Pacotes

Ser capaz de regular a forma do tráfego oferecido é um bom começo. However, para fornecer uma garantia de desempenho, devemos reservar recursos suficientes ao longo da rota que os pacotes fazem através da rede. Para fazer isso, somos as-supondo que os pacotes de um fluxo seguem a mesma rota. Pulverizando-os sobre a rota aleatoriamente torna difícil garantir qualquer coisa. Como consequência, algo semelhante a um circuito virtual deve ser configurado da fonte ao destino, e todos os pacotes que pertencem ao fluxo devem seguir esta rota.

Algoritmos que alocam recursos do roteador entre os pacotes de um fluxo e entre fluxos concorrentes são chamados **de algoritmos de programação de pacotes**. Três diferentes

tipos de recursos podem ser potencialmente reservados para diferentes fluxos:

1. Largura de banda.
2. Espaço de buffer.
3. Ciclos de CPU.

O primeiro, largura de banda, é o mais óbvio. Se um fluxo requer 1 Mbps e o linha de saída tem capacidade de 2 Mbps, tentando direcionar três fluxos através desse linha não vai funcionar. Assim, reservar largura de banda significa não sobrecarregar qualquer linha de saída.

Um segundo recurso que geralmente é insuficiente é o espaço de buffer. Quando um pacote chega, é armazenado em buffer dentro do roteador até que possa ser transmitido no linha de saída. O objetivo do buffer é absorver pequenas rajadas de tráfego como o

os fluxos competem uns com os outros. Se nenhum buffer estiver disponível, o pacote deve ser descartado, pois não há lugar para colocá-lo. Para uma boa qualidade de serviço, alguns buffers pode ser reservado para um fluxo específico para que o fluxo não tenha que competir por

412

A CAMADA DE REDE

INDIVÍDUO. 5

buffers com outros fluxos. Até algum valor máximo, sempre haverá um buffer disponível quando o fluxo precisa de um.

Finalmente, os ciclos da CPU também podem ser um recurso escasso. Leva tempo de CPU do roteador

para processar um pacote, então um roteador pode processar apenas um certo número de pacotes por segundo. Embora os roteadores modernos sejam capazes de processar a maioria dos pacotes rapidamente, alguns

tipos de pacotes requerem maior processamento da CPU, como os pacotes ICMP que irá descrever na Seç. 5.6. Certificar-se de que a CPU não está sobrecarregada é necessário para garantir o processamento atempado desses pacotes.

Algoritmos de programação de pacotes alocam largura de banda e outros recursos do roteador determinando qual dos pacotes em buffer enviar na próxima linha de saída. Nós já descreveu o planejador mais simples ao explicar como trabalhar. Cada roteador armazena pacotes em uma fila para cada linha de saída até que eles podem ser enviados e são enviados na mesma ordem em que chegaram. Este algoritmo é conhecido como **FIFO** (**First-In First-Out**), ou equivalentemente **FCFS** (**First-Come Primeiro serviço**).

Os roteadores FIFO geralmente descartam os pacotes recém-chegados quando a fila está cheia.

Como o pacote recém-chegado teria sido colocado no final da fila, esse comportamento é chamado de **queda da cauda** . É intuitivo, e você pode estar se perguntando o que

alternativas existem. Na verdade, o algoritmo RED que descrevemos na Seç. 5.3.5 escolheu um pacote recém-chegado para cair aleatoriamente quando o comprimento médio da fila cresceu ampla. Os outros algoritmos de programação que descreveremos também criam outras oportunidades para decidir qual pacote descartar quando os buffers estiverem cheios.

O agendamento FIFO é simples de implementar, mas não é adequado para fornecer boas qualidade de serviço, porque quando há vários fluxos, um fluxo pode facilmente afetar o desempenho dos outros fluxos. Se o primeiro fluxo é agressivo e envia grandes rajadas de pacotes, eles irão se alojar na fila. Processando pacotes no

ordem de sua chegada significa que o remetente agressivo pode monopolizar a maior parte da capacidade

dos roteadores que seus pacotes atravessam, privando os outros fluxos e reduzindo seus qualidade de serviço. Para adicionar insulto à injúria, os pacotes dos outros fluxos que fazem passar provavelmente será atrasado porque eles tiveram que sentar na fila atrás muitos pacotes do remetente agressivo.

Muitos algoritmos de programação de pacotes foram desenvolvidos para fornecer isolamento entre fluxos e impedir tentativas de interferência (Bhatti e Crowcroft, 2000). Um dos primeiros foi o algoritmo de **enfileiramento justo** desenvolvido por Nagle (1987). A essência deste algoritmo é que os roteadores têm filas separadas, uma para cada fluxo para uma determinada linha de saída. Quando a linha fica ociosa, o roteador verifica o round-robin das filas, conforme mostrado na Figura 5.30. Em seguida, ele pega o primeiro pacote no

próxima fila. Desta forma, com n hosts competindo pela linha de saída, cada host obtém para enviar um de cada n pacotes. É justo no sentido de que todos os fluxos são enviados pacotes na mesma taxa. O envio de mais pacotes não melhorará essa taxa.

Embora seja um começo, o algoritmo tem uma falha: ele dá mais largura de banda aos hosts que usam pacotes grandes do que hosts que usam pacotes pequenos. Demers et al. (1990) sugeriu uma melhoria em que o round robin é feito de forma a

SEC. 5,4
QUALIDADE DE SERVIÇO

413

Filas de entrada

Round-robin

serviço

1

2

3

1

12

3

2

3

Linha de saída

Figura 5-30. Filas justas de rodízio.

simular um round-robin byte a byte, em vez de um round robin pacote por pacote.

O truque é calcular um tempo virtual que é o número da rodada em que cada pacote terminaria de ser enviado. Cada rodada drena um byte de todos os filas que têm dados para enviar. Os pacotes são então classificados em ordem de suas finalidades horas de pesca e enviadas nessa ordem.

Este algoritmo é um exemplo de tempos de término para pacotes que chegam em três os fluxos são ilustrados na Figura 5-31. Se um pacote tiver comprimento L , a rodada em que terminará é simplesmente L voltas após a hora de início. O horário de início é hora do pacote anterior, ou a hora de chegada do pacote, se a fila for vazio quando chegar.

Filas de entrada

Justo

fila

Chegada do pacote

Tempo

Comprimento Final

Tempo

Resultado

ordem

UMA

0

8

8

1

B

5

6

11

3

C

5

10

10

2

D

8

9

20

7

E

8

8

14

4

F

10

6

16

5

G

11

10

19

6

H

20

8

28

8

UMA

B

C

```

E
G
D
F
H
Chega
tarde
(uma)
(b)
Chega depois de D
mas vai primeiro
O peso é 2
2X

```

Figura 5-31. (a) Enfileiramento justo ponderado. (b) Tempos de acabamento para os pacotes.

Da tabela da Fig. 5-32 (b), e olhando apenas para os primeiros dois pacotes na topo duas filas, os pacotes chegam de modo a $A, B, D, e F$. O pacote A chega em rodada 0 e tem 8 bytes de comprimento, então seu tempo de término é a rodada 8. Da mesma forma, o tempo de término para o pacote B é 11. O pacote D chega enquanto B está sendo enviado. Seu tempo de chegada é 9 byte-rounds depois de começar quando B termina, ou 20. Da mesma forma, o tempo de término de F é 16. Na ausência de novas chegadas, a ordem de envio relativa é A, B, F, D , mesmo embora F chegou após D . É possível que outro pequeno pacote chegue no topo fluir e obter um tempo de acabamento antes D . Ele só vai saltar à frente de D se o

Página 438

414

A CAMADA DE REDE INDIVÍDUO. 5

a transmissão desse pacote não foi iniciada. O enfileiramento justo não impede o pacote conjuntos que estão sendo transmitidos no momento. Como os pacotes são enviados em sua totalidade,

o enfileiramento justo é apenas uma aproximação do esquema byte a byte ideal. Mas isso é uma aproximação muito boa, ficando dentro de uma transmissão de pacote do ideal esquema em todos os momentos.

Uma lacuna deste algoritmo na prática é que ele dá a todos os hosts o mesma prioridade. Em muitas situações, é desejável fornecer, por exemplo, serviços de vídeo vers mais largura de banda do que, digamos, servidores de arquivos. Isso é facilmente possível, dando o

servidor de vídeo dois ou mais bytes por rodada. Este algoritmo modificado é chamado

WFQ (Weighted Fair Queuing). Deixando o número de bytes por rodada ser o peso de um fluxo, W , agora podemos fornecer a fórmula para calcular o tempo de término:

$$F_i = \max(A_i, F_{i-1}) + L_i / W$$

onde A_i é o tempo de chegada, F_i é o tempo de término e L_i é o comprimento do pacote i .

A fila inferior da Figura 5-31 (a) tem peso 2, de modo que seus pacotes são enviados mais rapidamente, como você pode ver nos tempos de término apresentados na Figura 5-31 (b).

Outra consideração prática é a complexidade da implementação. WFQ requer que os pacotes sejam inseridos por seu tempo de término em uma fila classificada. Com N fluxos, este

é, na melhor das hipóteses, uma operação $O(\log N)$ por pacote, o que é difícil de conseguir para muitos

fluxos em roteadores de alta velocidade. Shreedhar e Varghese (1995) descrevem um aproximação chamada **round robin de déficit**, que pode ser implementada de forma muito eficiente, com apenas $O(1)$ operações por pacote. WFQ é amplamente utilizado devido a esta aproximação.

Outros tipos de algoritmos de programação também existem. Um exemplo simples é a prioridade programação, em que cada pacote é marcado com uma prioridade. Pacotes de alta prioridade são sempre enviados antes de quaisquer pacotes de baixa prioridade armazenados em buffer. Dentro de um

ity, os pacotes são enviados em ordem FIFO. No entanto, o agendamento prioritário tem a desvantagem

vantagem de que uma explosão de pacotes de alta prioridade pode matar de fome pacotes de baixa prioridade, que

pode ter que esperar indefinidamente. O WFQ geralmente oferece uma alternativa melhor. Dando a fila de alta prioridade um grande peso, digamos 3, os pacotes de alta prioridade irão frequentemente através de uma linha curta (como relativamente poucos pacotes devem ser de alta prioridade), ainda alguns fração de pacotes de baixa prioridade continuarão a ser enviados mesmo quando houver alta tráfego prioritário. Um sistema de alta e baixa prioridade é essencialmente um WFQ de duas filas sistema no qual a alta prioridade tem peso infinito.

Como um exemplo final de um programador, os pacotes podem conter carimbos de data / hora e ser enviados na ordem do carimbo de data / hora. Clark et al. (1992) descreve um design em que o carimbo de data / hora registra o quanto o pacote está atrasado ou adiantado à medida que é enviado por meio de um sequência de roteadores no caminho. Pacotes que foram enfileirados atrás de outros pacotes em um roteador tendem a estar atrasados, e os pacotes que foram a manutenção primeiro tenderá a ser adiantada. Enviando pacotes na ordem de seus timestamps tem o efeito benéfico de acelerar pacotes lentos enquanto no ao mesmo tempo, diminuindo a velocidade dos pacotes. O resultado é que todos os pacotes são entregues pela rede com um atraso mais consistente.

Página 439

SEC. 5,4
QUALIDADE DE SERVIÇO

415

5.4.4 Controle de Admissão

Agora vimos todos os elementos necessários para QoS e é hora de colocar eles juntos para realmente fornecê-lo. As garantias de QoS são estabelecidas por meio do processo de controle de admissão. Vimos pela primeira vez o controle de admissão usado para controlar gestão, que é uma garantia de performance, ainda que fraca. As garantias que nós estão considerando agora são mais fortes, mas o modelo é o mesmo. O usuário oferece um fluxo com um requisito de QoS para a rede. A rede então decide se aceita ou rejeita o fluxo com base em sua capacidade e o compromisso mentos que fez a outros fluxos. Se aceitar, a rede reserva capacidade em avance nos roteadores para garantir QoS quando o tráfego for enviado no novo fluxo. As reservas devem ser feitas em todos os roteadores ao longo da rota que os pacotes passam pela rede. Quaisquer roteadores no caminho sem reservas pode ficar congestionado e um único roteador congestionado pode quebrar a garantia de QoS tee. Muitos algoritmos de roteamento encontram o melhor caminho único entre cada fonte e cada destino e enviar todo o tráfego pelo melhor caminho. Isso pode causar alguns fluxos a serem rejeitados se não houver capacidade sobressalente suficiente ao longo do melhor caminho. QoS

garantias para novos fluxos ainda podem ser acomodados escolhendo um rota para o fluxo que tem excesso de capacidade. Isso é chamado de **roteamento QoS**. Chen e Nahrstedt (1998) dá uma visão geral dessas técnicas. Também é possível dividir o tráfego para cada destino em vários caminhos para encontrar mais facilmente o excesso de capacidade. Um método simples é para os roteadores escolherem caminhos de custo igual e dividirem o tráfego igualmente ou em proporção à capacidade dos links de saída. Contudo, algoritmos mais sofisticados também estão disponíveis (Nelakuditi e Zhang, 2002). Dado um caminho, a decisão de aceitar ou rejeitar um fluxo não é uma simples questão de comparar os recursos (largura de banda, buffers, ciclos) solicitados pelo fluxo com o excesso de capacidade do roteador nessas três dimensões. É um pouco mais complicatedo do que isso. Para começar, embora alguns aplicativos possam saber sobre seus requisitos de largura de banda, poucos sabem sobre buffers ou ciclos de CPU, portanto, no mini-mãe, uma maneira diferente é necessária para descrever os fluxos e traduzir esta descrição para recursos do roteador. Chegaremos a isso em breve.

Em seguida, algumas aplicações são muito mais tolerantes a perdas ocasionais de mortos linha do que outros. Os aplicativos devem escolher o tipo de garantias que que a rede pode fazer, sejam garantias rígidas ou comportamento que manterá a maior parte A Hora. Tudo o mais sendo igual, todos gostariam de garantias rígidas, mas a dificuldade A verdade é que eles são caros porque restringem o comportamento do pior caso. Garantees para a maioria dos pacotes são frequentemente suficientes para aplicativos e mais fluxos com esta garantia pode ser suportado por uma capacidade fixa.

Finalmente, alguns aplicativos podem estar dispostos a pechinchar sobre os parâmetros de fluxo e outros não. Por exemplo, um visualizador de filme que normalmente roda em 30 quadros / s podem estar dispostos a voltar para 25 quadros / s se não houver o suficiente largura de banda livre para suportar 30 frames / seg. Da mesma forma, o número de pixels por quadro, largura de banda de áudio e outras propriedades podem ser ajustáveis.

Página 440

416

A CAMADA DE REDE INDIVÍDUO. 5

Porque muitas partes podem estar envolvidas na negociação do fluxo (o remetente, o receptor, e todos os roteadores ao longo do caminho entre eles), os fluxos devem ser de-escritas com precisão em termos de parâmetros específicos que podem ser negociados. Um conjunto de

esses parâmetros são chamados de **especificação de fluxo**. Normalmente, o remetente (por exemplo, o servidor de vídeo) produz uma especificação de fluxo propondo os parâmetros que deseja usar. Conforme a especificação se propaga ao longo da rota, cada roteador a examina e modifica os parâmetros conforme necessário. As modificações podem apenas reduzir o fluxo, não aumente (por exemplo, uma taxa de dados mais baixa, não uma mais alta). Quando chega ao

outra extremidade, os parâmetros podem ser estabelecidos.

Como um exemplo do que pode estar em uma especificação de fluxo, considere o exemplo de Fig. 5-32. Isso é baseado nas RFCs 2210 e 2211 para serviços integrados, um QoS design que iremos cobrir na próxima seção. Possui cinco parâmetros. Os dois primeiros pa-

parâmetros, a *taxa de balde de símbolos* e *tamanho balde de símbolos*, usar um balde de símbolos para

dar

a taxa máxima sustentada que o remetente pode transmitir, calculada ao longo de um longo tempo intervalo, e o maior burst que pode enviar em um curto intervalo de tempo.

Parâmetro

Unidade

Taxa de depósito de token

Bytes / s

Tamanho do balde de token

Bytes

Taxa de pico de dados

Bytes / s

Tamanho mínimo do pacote

Bytes

Tamanho máximo do pacote

Bytes

Figura 5-32. Um exemplo de especificação de fluxo.

O terceiro parâmetro, a *taxa de pico de dados*, é a taxa de transmissão máxima tolerado, mesmo por breves intervalos de tempo. O remetente nunca deve exceder esta taxa mesmo para rajadas curtas.

Os dois últimos parâmetros especificam os tamanhos de pacote mínimo e máximo, em incluindo os cabeçalhos das camadas de transporte e rede (por exemplo, TCP e IP). O mínimo tamanho é útil porque o processamento de cada pacote leva algum tempo fixo, não importa quanto curto. Um roteador pode ser preparado para lidar com 10.000 pacotes / s de 1 KB cada, mas não esteja preparado para lidar com 100.000 pacotes / s de 50 bytes cada, embora isso represente uma taxa de dados mais baixa. O tamanho máximo do pacote é importante devido a limitações da rede interna que não podem ser excedidas. Por exemplo, se parte do

caminho passa por uma Ethernet, o tamanho máximo do pacote será restrito a não mais de 1500 bytes, não importa o que o resto da rede possa suportar.

Uma questão interessante é como um roteador transforma uma especificação de fluxo em um conjunto de

reservas de recursos específicos. À primeira vista, pode parecer que se um roteador tiver um link que roda a, digamos, 1 Gbps e o pacote médio é de 1000 bits, ele pode processar 1 milhão de pacotes / s. Esta observação não é o caso, porque haverá

Sempre haverá períodos ociosos no link devido a flutuações estatísticas na carga. Se o

Página 441

SEC. 5,4

QUALIDADE DE SERVIÇO

417

o link precisa de cada bit de capacidade para fazer seu trabalho, parando até mesmo por alguns bits cria um acúmulo do qual nunca consegue se livrar.

Mesmo com uma carga um pouco abaixo da capacidade teórica, as filas podem se acumular e atrasos podem ocorrer. Considere uma situação em que os pacotes chegam aleatoriamente com uma taxa média de chegada de λ pacotes / s. Os pacotes têm comprimentos aleatórios e podem ser enviado no link com uma taxa média de serviço de pacotes μ / s. Sob a suposição que ambas as distribuições de chegada e serviço são distribuições de Poisson (o que é chamado de sistema de filas M / M / 1, onde "M" representa Markov, ou seja, Poisson), pode ser comprovado usando a teoria das filas que o atraso médio experimentado por um pacote, T , é

$$T =$$

$$\mu$$

$$1$$

×

$$1 - \lambda / \mu$$

$$1$$

=

$$\mu$$

$$1$$

×

$$1 - \rho$$

$$1$$

onde $\rho = \lambda / \mu$ é a utilização da CPU. O primeiro fator, $1 / \mu$, é o que o serviço tempo seria na ausência de competição. O segundo fator é a desaceleração devido à competição com outros fluxos. Por exemplo, se $\lambda = 950.000$ pacotes / s e $\mu = 1.000.000$ pacotes / s, então $\rho = 0,95$ e o atraso médio experimentado por cada o pacote será de 20 μ s em vez de 1 μ s. Desta vez, tanto a fila o tempo de serviço e o tempo de serviço, como pode ser observado quando a carga é muito baixa ($\lambda / \mu \sim 0$).

Se houver, digamos, 30 roteadores ao longo da rota do fluxo, apenas o atraso da fila irá agir conte para 600 μ s de atraso.

Um método de relacionar as especificações de fluxo aos recursos do roteador que correspondem a largura de banda e as garantias de desempenho de atraso são fornecidas por Parekh e Gallagher (1993, 1994). É baseado em fontes de tráfego moldadas por (R, B) token buckets e WFQ em roteadores. Cada fluxo recebe um peso WFQ W grande o suficiente para drenar seu taxa de token bucket R , conforme mostrado na Fig. 5-33. Por exemplo, se o fluxo tem uma taxa de 1 Mbps e o roteador e link de saída têm uma capacidade de 1 Gbps, o peso para o o fluxo deve ser maior que 1/1000 do total dos pesos para todos os fluxos nesse roteador para o link de saída. Isso garante ao fluxo uma largura de banda mínima. Se não puder ser fornecida uma taxa grande o suficiente, o fluxo não pode ser admitido.

Pesada
fila justa

(R, B)

Fonte de tráfego

Roteador

Capacidade C

W
W_{eu}
W_{eu}
WxC
R <
pesos

Figura 5-33. Largura de banda e garantias de atraso com token buckets e WFQ.

O maior atraso na fila que o fluxo verá é uma função do tamanho do burst de o token bucket. Considere os dois casos extremos. Se o trânsito estiver bom, sem

418

A CAMADA DE REDE

INDIVÍDUO. 5

quaisquer bursts, os pacotes serão drenados do roteador assim que chegarem. Não haverá atraso na fila (ignorando os efeitos de empacotamento). No outro Por outro lado, se o tráfego é salvo em rajadas, então uma rajada de tamanho máximo, B , pode chegar no roteador de uma vez. Neste caso, o atraso máximo de enfileiramento, D , será o tempo necessário para drenar este burst na largura de banda garantida, ou B/R (novamente, ignorando efeitos de empacotamento). Se este atraso for muito grande, o fluxo deve solicitar mais banda largura da rede.

Essas garantias são difíceis. Os balde de tokens limitam a explosão do a origem e o enfileiramento justo isolam a largura de banda fornecida a diferentes fluxos. Isto significa que o fluxo atenderá suas garantias de largura de banda e atraso, independentemente de como os outros fluxos concorrentes se comportam no roteador. Esses outros fluxos não podem quebre a garantia economizando tráfego e enviando tudo de uma vez.

Além disso, o resultado é válido para um caminho através de vários roteadores em qualquer rede topologia. Cada fluxo obtém uma largura de banda mínima porque essa largura de banda é garantida instalado em cada roteador. A razão pela qual cada fluxo obtém um atraso máximo é mais subtle. No pior caso em que uma explosão de tráfego atinge o primeiro roteador e compete com o tráfego de outros fluxos, ele será atrasado até ao atraso máximo de D . Como-nunca, esse atraso também vai suavizar o estouro. Por sua vez, isso significa que a explosão irá não incorrer em mais atrasos na fila em roteadores posteriores. O atraso geral na fila irá ser no máximo D .

5.4.5 Serviços Integrados

Entre 1995 e 1997, o IETF se esforçou muito para criar uma arquitetura para streaming de multimídia. Este trabalho resultou em mais de duas dúzias de RFCs, começando com RFCs 2205–2212. O nome genérico para este trabalho é **serviços integrados**. isto foi direcionado para aplicações unicast e multicast. Um exemplo do primeiro é um único usuário transmitindo um videoclipe de um site de notícias. Um exemplo deste último é uma coleção de estações de televisão digital transmitindo seus programas como streams de pacotes IP para muitos receptores em vários locais. Abaixo vamos nos concentrar no multicast, uma vez que o unicast é um caso especial de multicast.

Em muitos aplicativos multicast, os grupos podem alterar a associação dinamicamente, por exemplo, quando as pessoas entram em uma videoconferência e ficam entediadas e mudam para uma novela ou o canal de croquet. Nessas condições, a abordagem de hav-fazer com que os remetentes reservem largura de banda com antecedência não funciona bem, pois exigem que cada remetente rastreie todas as entradas e saídas de seu público. Para um sistema de de-assinado para transmitir televisão com milhões de assinantes, não funcionaria de todo.

RSVP - O protocolo de reSerVação de recursos

A parte principal da arquitetura de serviços integrados que é visível para os usuários da rede é **RSVP**. Ele é descrito nas RFCs 2205–2210. Este protocolo é utilizado para fazer as reservas; outros protocolos são usados para enviar os dados.

RSVP permite que vários remetentes transmitam para vários grupos de receptores, permite receptores individuais para mudar de canal livremente e otimizar o uso de largura de banda enquanto ao mesmo tempo, eliminando o congestionamento.

Em sua forma mais simples, o protocolo usa roteamento multicast usando árvores de abrangência, Como discutido anteriormente. Cada grupo recebe um endereço de grupo. Para enviar para um grupo,

um remetente coloca o endereço do grupo em seus pacotes. O roteamento multicast padrão também ggorithm então constrói uma árvore abrangente cobrindo todos os membros do grupo. O roteamento al-

ggorithm não faz parte do RSVP. A única diferença do multicast normal é um pouca informação extra que é multicast para o grupo periodicamente para dizer aos roteadores ao longo da árvore para manter certas estruturas de dados em suas memórias.

Como exemplo, considere a rede da Figura 5.34 (a). Hosts 1 e 2 são multi-remetentes de transmissão e os hosts 3, 4 e 5 são receptores de multicast. Neste exemplo, os remetentes e receptores são separados, mas, em geral, os dois conjuntos podem se sobrepor. o árvores multicast para os hosts 1 e 2 são mostradas na Fig. 5-34 (b) e na Fig. 5-34 (c), respectivamente.

```

UMA
D
G
J
C
F
Eu
eu
B
K
H
E
1
2
3
4
5
Receptores
Remetentes
UMA
D
G
J
C
F
Eu
eu
B
K
H
E
1
2
3
4
5
1
2
3
4
5
UMA
D
G
J
C
F
Eu
eu
B
K
H
E
(uma)
(b)
(c)

```

Figura 5-34. (a) Uma rede. (b) A árvore de abrangência multicast para o host 1. (c) O árvore de abrangência multicast para o host 2.

Para obter uma melhor recepção e eliminar o congestionamento, qualquer um dos receptores em um O grupo pode enviar uma mensagem de reserva até o remetente. A mensagem é propagado usando o algoritmo de encaminhamento de caminho reverso discutido anteriormente. Em cada

420

A CAMADA DE REDE

INDIVÍDUO. 5

salto, o roteador anota a reserva e reserva a largura de banda necessária. Nós vimos na seção anterior como um planejador de enfileiramento justo ponderado pode ser usado para fazer esta reserva. Se houver largura de banda insuficiente, ele reporta de volta falha. No momento em que a mensagem volta para a fonte, a largura de banda foi refeita servido desde o remetente até o destinatário que faz a solicitação de reserva ao longo da árvore de abrangência.

Um exemplo de tal reserva é mostrado na Figura 5-35 (a). Aqui o host 3 tem solicitou um canal para hospedar 1. Uma vez estabelecido, os pacotes podem fluir de 1 a 3 sem congestionamento. Agora, considere o que acontece se o host 3 em seguida reserva um canal para o outro remetente, host 2, para que o usuário possa assistir dois programas de televisão

programas de uma vez. Um segundo caminho está reservado, conforme ilustrado na Figura 5-35 (b). Nota

que dois canais separados são necessários do host 3 ao roteador *E* porque dois fluxos pendentes estão sendo transmitidos.

```

UMA
D
G
J
C
F
Largura de banda reservada
para fonte 1
Largura de banda
reservado para
fonte 2
Eu
eu
B
K
H
E
1
2
3
4
5
UMA
D
G
J
C
F
Eu
eu
B
K
H
E
1
2
2
3
4
5
UMA
D
G
J
C
F
Eu
eu
B
K
H
E
(b)
(c)
(uma)
3
4
5
1

```

Figura 5-35. (a) Host 3 solicita um canal para o host 1. (b) Host 3 então solicita um segundo canal, para o host 2. (c) O host 5 solicita um canal para o host 1.

Finalmente, na Figura 5-35 (c), o host 5 decide assistir ao programa sendo transmitido

pelo host 1 e também faz uma reserva. Primeiro, a largura de banda dedicada é reservada como Tanto quanto roteador H . No entanto, este roteador vê que já tem um feed do host 1, então se a largura de banda necessária já foi reservada, não é necessário reservar não mais. Observe que os hosts 3 e 5 podem ter solicitado diferentes quantidades de largura de banda (por exemplo, se o host 3 está jogando em uma tela pequena e só deseja o informações de resolução), de modo que a capacidade reservada deve ser grande o suficiente para satisfazer

o receptor mais ganancioso.

Ao fazer uma reserva, um receptor pode (opcionalmente) especificar um ou mais fontes das quais deseja receber. Ele também pode especificar se essas escolhas

Página 445

SEC. 5,4
QUALIDADE DE SERVIÇO

421

são fixos para a duração da reserva ou se o destinatário deseja manter abra a opção de alterar as fontes mais tarde. Os roteadores usam essas informações para operar otimizar o planejamento da largura de banda. Em particular, dois receptores são configurados apenas para compartilhar um

caminho se ambos concordarem em não alterar as fontes posteriormente.

A razão para esta estratégia no caso totalmente dinâmico é que a banda reservada largura é desacoplada da escolha da fonte. Assim que o receptor reservou banda-largura, ele pode mudar para outra fonte e manter essa parte do caminho existente isso é válido para a nova fonte. Se o host 2 estiver transmitindo vários streams de vídeo em tempo real, por exemplo, uma emissora de TV com vários canais, o host 3 pode alternar entre eles à vontade, sem alterar sua reserva: os roteadores não importa qual programa o receptor está assistindo.

5.4.6 Serviços Diferenciados

Algoritmos baseados em fluxo têm o potencial de oferecer boa qualidade de serviço para um ou mais fluxos porque eles reservam todos os recursos necessários ao longo do rota. No entanto, eles também têm uma desvantagem. Eles exigem uma configuração avançada para es-

tabelecer cada fluxo, algo que não se ajusta bem quando há milhares ou milhões de fluxos. Além disso, eles mantêm o estado interno por fluxo nos roteadores, mak- tornando-os vulneráveis a travamentos do roteador. Finalmente, as mudanças necessárias para o roteador

código são substanciais e envolvem trocas de roteador para roteador complexas para configurar os fluxos. Como consequência, enquanto o trabalho continua para o avanço de serviços integrados víscios, poucas implantações ou algo parecido ainda existe.

Por essas razões, a IETF também desenvolveu uma abordagem mais simples para a qualidade do serviço

vice, aquele que pode ser amplamente implementado localmente em cada roteador sem avanço configuração e sem ter todo o caminho envolvido. Esta abordagem é conhecida como qualidade de serviço **baseada em classe** (em oposição a baseada em fluxo). IETF padronizou uma arquitetura para ele, chamada de **serviços diferenciados**, que é descrita em RFCs 2474, 2475 e vários outros. Vamos agora descrevê-lo.

Serviços diferenciados podem ser oferecidos por um conjunto de roteadores formando um administrador

domínio trativo (por exemplo, um ISP ou uma telco). A administração define um conjunto de serviços

classes com regras de encaminhamento correspondentes. Se um cliente assina dif- serviços diferenciados, os pacotes do cliente que entram no domínio são marcados com o classe a que pertencem. Esta informação é transportada no *Serviço Diferenciado* campo *víscios* de pacotes IPv4 e IPv6 (descritos na Seção 5.6). As aulas são de- multado de acordo com os comportamentos de salto porque correspondem ao tratamento do pacote

receberá em cada roteador, não uma garantia em toda a rede. Melhor serviço é

fornecido para pacotes com alguns comportamentos por salto (por exemplo, serviço premium) do que para outros (por exemplo, serviço regular). O tráfego dentro de uma classe pode ser necessário para estar em conformidade com alguma forma específica, como um balde com vazamento com alguma taxa de drenagem especificada. A operador com faro para negócios pode cobrar mais por cada pacote premium transportado ou pode permitir até N pacotes premium por mês para um suplemento fixo taxa mensal internacional. Observe que este esquema não requer configuração prévia, nenhum recurso

Página 446

422

A CAMADA DE REDE INDIVÍDUO. 5

reserva, e nenhuma negociação de ponta a ponta demorada para cada fluxo, como acontece com serviços integrados. Isso torna serviços diferenciados relativamente fáceis de implementar ment.

O serviço baseado em classe também ocorre em outros setores. Por exemplo, pacote de- as empresas de pintura geralmente oferecem serviço noturno, de dois ou três dias. Companhias aéreas

oferecem serviço de primeira classe, classe executiva e classe de gado. Trens de longa distância frequentemente têm várias classes de serviço. Até o metrô de Paris tem dois serviços diferentes vice classes. Para pacotes, as classes podem diferir em termos de atraso, jitter e probabilidade de ser descartado em caso de congestionamento, entre outras possibilidades laços (mas provavelmente não frames Ethernet mais espaçoso).

Para fazer a diferença entre qualidade de serviço baseada em fluxo e qualidade de serviço baseada em classe

qualidade de serviço mais clara, considere um exemplo: telefonia pela Internet. Com um fluxo esquema baseado, cada chamada telefônica tem seus próprios recursos e garantias. Com um esquema baseado em classe, todas as chamadas telefônicas juntas obtêm os recursos reservados para a classe de telefonia. Esses recursos não podem ser retirados por pacotes do

Aulas de navegação na web ou outras aulas, mas nenhuma chamada telefônica obtém qualquer resposta privada

fontes reservadas apenas para ele.

Encaminhamento acelerado

A escolha das classes de serviço é de cada operadora, mas desde que os pacotes sejam frequentemente encaminhado entre redes operadas por diferentes operadoras, a IETF definiu algumas classes de serviço independentes da rede. A classe mais simples é **acelerada para-guarda**, então vamos começar com aquele. Ele é descrito no RFC 3246.

A ideia por trás do encaminhamento acelerado é muito simples. Duas classes de serviço estão disponíveis: regulares e acelerados. A grande maioria do tráfego é esperada para ser regular, mas uma fração limitada dos pacotes é acelerada. O acelerado os pacotes devem ser capazes de transitar pela rede como se nenhum outro pacote fosse presente. Desta forma, eles obterão baixa perda, baixo atraso e serviço de baixo jitter - apenas o que é necessário para VoIP. Uma representação simbólica deste sistema de "dois tubos" é dado na Figura 5-36. Observe que ainda há apenas uma linha física. Os dois lógicos os tubos mostrados na figura representam uma maneira de reservar largura de banda para diferentes classes de serviço, não uma segunda linha física.

Uma maneira de implementar essa estratégia é a seguinte. Os pacotes são classificados como acelerado ou regular e marcado em conformidade. Esta etapa pode ser realizada no host de envio ou no roteador de entrada (primeiro). A vantagem de fazer classificação no host de envio é que mais informações estão disponíveis sobre quais pacotes serão longo para o qual flui. Esta tarefa pode ser realizada por software de rede ou mesmo o sistema operacional, para evitar a necessidade de alterar os aplicativos existentes. Para ex- amplamente, está se tornando comum que os pacotes de VoIP sejam marcados para serviço expresso vice pelos anfitriões. Se os pacotes passarem por uma rede corporativa ou ISP que ofereça

no atendimento acelerado dos portos, eles receberão tratamento preferencial. Se a rede não suporta serviço expresso, nenhum dano é causado.

Página 447

SEC. 5,4

QUALIDADE DE SERVIÇO

423

Pacotes regulares

Pacotes acelerados

Figura 5-36. Os pacotes acelerados experimentam uma rede sem tráfego.

Claro, se a marcação for feita pelo host, o roteador de ingresso provavelmente policiar o tráfego para garantir que os clientes não enviem tráfego mais rápido do que eles pagaram. Dentro da rede, os roteadores podem ter duas saídas filas para cada linha de saída, uma para pacotes expedidos e outra para pacotes regulares etc. Quando um pacote chega, ele é enfileirado de acordo. A fila acelerada é dada prioridade sobre o normal, por exemplo, usando um agendador de prioridade. No desta forma, os pacotes expedidos veem uma rede descarregada, mesmo quando há, de fato, uma carga pesada de tráfego regular.

Encaminhamento garantido

Um esquema um pouco mais elaborado para gerenciar as classes de serviço é chamado **encaminhamento garantido**. É descrito no RFC 2597. Encaminhamento garantido especifica que deve haver quatro classes prioritárias, cada classe com seus próprios recursos. O as três classes principais podem ser chamadas de ouro, prata e bronze. Além disso, define três classes de descarte para pacotes que estão enfrentando congestionamento: baixo, médio, e alto. Juntos, esses dois fatores definem 12 classes de serviço.

A Figura 5-37 mostra que os pacotes de uma via podem ser processados sob garantia para-guarda. O primeiro passo é classificar os pacotes em uma das quatro prioridades Aulas. Como antes, esta etapa pode ser realizada no host de envio (conforme mostrado no figura) ou no roteador de ingresso, e a taxa de pacotes de prioridade mais alta pode ser limitada itado pela operadora como parte da oferta de serviço.

A próxima etapa é determinar a classe de descarte de cada pacote. Isso está feito passando os pacotes de cada classe de prioridade por meio de um vigilante de tráfego, como um balde de tokens. O vigilante deixa todo o tráfego passar, mas identifica os pacotes que cabem em pequenas rajadas como descarte baixo, pacotes que excedem pequenas rajadas como descarte médio e pacotes que excedem grandes bursts como descarte alto. O combinação de prioridade e classe de descarte é então codificada em cada pacote.

Finalmente, os pacotes são processados por roteadores na rede com um pacote programador que distingue as diferentes classes. Uma escolha comum é usar

Página 448

424

A CAMADA DE REDE

INDIVÍDUO. 5

Pesada

filas justas

Roteador

Prata

Ouro

Bronze

Pacote

fente

Quatro

prioridade

Aulas

Classificador

Policer

Doze

prioridade / descartar

Aulas

Pacotes com

Marca DiffServ

Figura 5-37. Uma possível implementação de encaminhamento garantido.

enfileiramento justo ponderado para as quatro classes prioritárias, com classes mais altas dadas

pesos mais elevados. Desta forma, as classes mais altas obterão a maior parte da largura de banda, mas

as classes mais baixas não ficarão totalmente sem largura de banda. Por exemplo, se o pesos dobram de uma classe para a próxima classe superior, a classe superior obterá duas vezes a largura de banda. Dentro de uma classe de prioridade, os pacotes com uma classe de descarte mais alta

pode ser preferencialmente descartado executando um algoritmo como RED (Random Detecção Antecipada), que vimos na Seç. 5.3.5. RED começará a descartar pacotes como o congestionamento aumenta, mas antes que o roteador fique sem espaço de buffer. Nesta fase, ainda há espaço de buffer com o qual aceitar pacotes de descarte baixo enquanto descarta pacotes de descarte alto.

5.5 INTERNETWORKING

Até agora, assumimos implicitamente que existe um único homogêneo rede, com cada máquina usando o mesmo protocolo em cada camada. Infelizmente, essa suposição é extremamente otimista. Existem muitas redes diferentes, incluindo PANs, LANs, MANs e WANs. Descrevemos Ethernet, Internet over cabo, as redes de telefonia fixa e móvel, 802.11, 802.16 e muito mais. Num-vários protocolos são amplamente usados por essas redes em todas as camadas. No nas próximas seções, daremos uma olhada cuidadosa nos problemas que surgem quando dois ou mais redes são conectadas para formar uma **internetwork**, ou mais simplesmente uma **internet**.

Seria muito mais simples unir redes se todos usassem um único tecnologia de rede, e muitas vezes é o caso de haver um tipo dominante de rede, como Ethernet. Alguns especialistas especulam que a multiplicidade de tecnologias ogies irão embora assim que todos perceberem o quanto maravilhoso [preencha seu favor-rede ite] é. Não conte com isso. A história mostra que isso é ilusão. Dif-diferentes tipos de redes lidam com diferentes problemas, então, por exemplo, Ether-é provável que as redes de rede e de satélite sejam sempre diferentes. Reutilizando sistemas existentes, como a execução de redes de dados por cabo, a rede telefônica e energia

Página 449

SEC. 5,5
INTERNETWORKING

425

linhas, adiciona restrições que fazem com que os recursos das redes divergam. Hetero-a geneidade está aqui para ficar.

Se sempre haverá redes diferentes, seria mais simples se não precisa interconectá-los. Isso também é improvável. Bob Metcalfe postulou que o valor de uma rede com N nós é o número de conexões que podem ser feitas entre os nós, ou N^2 (Gilder, 1993). Isso significa que grandes redes são muito mais valioso do que pequenas redes porque permitem muito mais conexões, então sempre haverá um incentivo para combinar redes menores.

A Internet é o principal exemplo dessa interconexão. (Vamos escrever em ternet com um "I" maiúsculo para distingui-lo de outras internets, ou rede conectada-funciona.) O objetivo de unir todas essas redes é permitir que os usuários em qualquer um dos para que eles se comuniquem com usuários em todos os outros. Quando você paga um ISP por Serviço de Internet, você pode ser cobrado dependendo da largura de banda de sua linha, mas o que você está realmente pagando é a capacidade de trocar pacotes com qualquer outro host que também está conectado à Internet. Afinal, a Internet não seria muito popular se você pudesse apenas enviar pacotes para outros hosts na mesma cidade.

Uma vez que as redes muitas vezes diferem em aspectos importantes, obter pacotes de uma rede trabalhar para outra nem sempre é tão fácil. Devemos abordar os problemas de hetero-geneidade e também problemas de escala à medida que a Internet resultante cresce muito. Nós começaremos observando como as redes podem diferir para ver contra o que estamos lutando.

Então veremos a abordagem usada com tanto sucesso pelo IP (Internet Protocol), o protocolo da camada de rede da Internet, incluindo técnicas para tunelamento através

redes, roteamento em internetworks e fragmentação de pacotes.

5.5.1 Como as redes diferem

As redes podem diferir de várias maneiras. Algumas das diferenças, como diferentes técnicas de modulação ou formatos de quadro, são internos ao link físico e de dados camadas. Essas diferenças não nos preocupam aqui. Em vez disso, na Figura 5-38, listamos algumas das diferenças que podem ser expostas à camada de rede. É papel sobre essas diferenças que tornam a internetworking mais difícil do que operar dentro de uma única rede.

Quando os pacotes enviados por uma fonte em uma rede devem transitar por um ou mais para-redes externas antes de chegar à rede de destino, muitos problemas podem ocorrer nas interfaces entre as redes. Para começar, a fonte precisa ser capaz de endereçar o destino. O que faremos se a fonte estiver em uma rede Ethernet e o destino está em uma rede WiMAX? Supondo que possamos até mesmo especificar um Destino WiMAX de uma rede Ethernet, os pacotes cruzariam de uma rede sem conexão para uma orientada para conexão. Isso pode exigir que um novo conexão ser configurada em curto prazo, o que injeta um atraso e muito overhead se a conexão não é usada para muitos mais pacotes.

Muitas diferenças específicas também devem ser acomodadas. Como nós multicast um pacote para um grupo com alguns membros em uma rede que não

Página 450

426

A CAMADA DE REDE

INDIVÍDUO. 5

Item

Algumas possibilidades

Serviço oferecido

Sem conexão versus orientado para conexão

Endereçando

Tamanhos diferentes, planos ou hierárquicos

Transmissão

Presente ou ausente (também multicast)

Tamanho do pacote

Cada rede tem seu próprio máximo

Encomenda

Entrega ordenada e não ordenada

Qualidade de serviço

Presente ou ausente; muitos tipos diferentes

Confiabilidade

Diferentes níveis de perda

Segurança

Regras de privacidade, criptografia, etc.

Parâmetros

Diferentes tempos limite, especificações de fluxo, etc.

Contabilidade

Por tempo de conexão, pacote, byte ou nada

Figura 5-38. Algumas das muitas maneiras pelas quais as redes podem ser diferentes.

suporta multicast? Os diferentes tamanhos máximos de pacotes usados por diferentes redes podem ser um grande incômodo também. Como você passa um pacote de 8000 bytes através de uma rede trabalho cujo tamanho máximo é de 1500 bytes? Se os pacotes em uma conexão orientada transitar por uma rede sem conexão, eles podem chegar em uma ordem diferente da que eles foram enviados. Isso é algo que o remetente provavelmente não esperava, e pode vir como uma (desagradável) surpresa para o receptor também.

Esses tipos de diferenças podem ser ocultados, com algum esforço. Para exemplo, um gateway unindo duas redes pode gerar pacotes separados para cada destino em vez de melhor suporte de rede para multicast. Um pacote grande pode ser quebrado, enviado em pedaços e, em seguida, reunido novamente. Receptores podem armazenar pacotes em buffer e entregá-los em ordem.

As redes também podem diferir em grandes aspectos que são mais difíceis de reconciliar. O exemplo mais claro é a qualidade do serviço. Se uma rede tiver QoS forte e o outro oferece melhor serviço de esforço, será impossível fazer largura de banda e atrasar

garantias de tráfego em tempo real de ponta a ponta. Na verdade, eles provavelmente só podem ser feitos

enquanto a rede de melhor esforço é operada com baixa utilização, ou pouco usada, o que é improvável que seja o objetivo da maioria dos ISPs. Mecanismos de segurança são problemáticos, mas pelo menos a criptografia para confidencialidade e integridade de dados pode ser implementada em cima de redes que ainda não o incluem. Finalmente, as diferenças em a-contagem pode levar a contas indesejadas quando o uso normal de repente se torna ex-pensativo, como os usuários de roaming de telefones móveis com planos de dados descobriram.

5.5.2 Como as redes podem ser conectadas

Existem duas opções básicas para conectar redes diferentes: podemos construir dispositivos que traduzem ou convertem pacotes de cada tipo de rede em pacotes para cada outra rede, ou, como bons cientistas da computação, podemos tentar resolver o

Página 451

SEC. 5,5
INTERNETWORKING

427

problema, adicionando uma camada de indireção e construindo uma camada comum em cima de as diferentes redes. Em qualquer caso, os dispositivos são colocados nos limites entre redes interpoladas.

No início, Cerf e Kahn (1974) defenderam uma camada comum para esconder a diferença de diferenças de redes existentes. Essa abordagem tem sido um tremendo sucesso, e a camada que eles propuseram foi eventualmente separada no protocolo TCP e IP cols. Quase quatro décadas depois, o IP é a base da Internet moderna. Para esta conquista, Cerf e Kahn receberam o Prêmio Turing de 2004, mal conhecido como o Prêmio Nobel da ciência da computação. IP fornece um universal formato de pacote que todos os roteadores reconhecem e que pode ser transmitido quase cada rede. IP estendeu seu alcance de redes de computadores para assumir a rede telefônica. Ele também funciona em redes de sensores e outros dispositivos minúsculos que foram considerados muito limitados em recursos para suportá-lo.

Discutimos vários dispositivos diferentes que conectam redes, incluindo repetidores, hubs, switches, bridges, roteadores e gateways. Repetidores e hubs apenas mover bits de um fio para outro. Eles são principalmente dispositivos analógicos e não entender qualquer coisa sobre protocolos de camada superior. Pontes e interruptores operam em a camada de link. Eles podem ser usados para construir redes, mas apenas com protocolo secundário tradução no processo, por exemplo, entre 10, 100 e 1000 Mbps Ethernet comuta. Nossa foco nesta seção são os dispositivos de interconexão que operam no camada de rede, ou seja, os roteadores. Vamos deixar os gateways, que são mais altos dispositivos de interconexão de camadas, até mais tarde.

Vamos primeiro explorar em um alto nível como a interconexão com uma rede comum a camada de trabalho pode ser usada para interconectar redes diferentes. Uma internet composta de redes 802.11, MPLS e Ethernet é mostrada na Figura 5.39 (a).

Suponha que a máquina de origem na rede 802.11 deseja enviar um pacote para a máquina de destino na rede Ethernet. Uma vez que essas tecnologias são diferentes diferentes, e eles são separados por outro tipo de rede (MPLS), alguns processamento adicional é necessário nas fronteiras entre as redes.

Porque diferentes redes podem, em geral, ter diferentes formas de anúncio vestir, o pacote carrega um endereço de camada de rede que pode identificar qual host a-cruzar as três redes. O primeiro limite que o pacote atinge é quando ele transições de uma rede 802.11 para uma rede MPLS. 802.11 fornece uma conexão sem conexão, mas MPLS fornece um serviço orientado a conexão. Isso significa que um circuito virtual deve ser configurado para cruzar essa rede. Assim que o pacote tiver percorrido ao longo do circuito virtual, ele alcançará a rede Ethernet. Neste limite, o pacote pode ser muito grande para ser transportado, uma vez que 802.11 pode funcionar com quadros maiores do que Ethernet. Para lidar com este problema, o pacote é dividido em fragmentos, e cada fragmento é enviado separadamente. Quando os fragmentos alcançam o

destino, eles são remontados. Então, o pacote completou sua jornada.

O processamento do protocolo para essa jornada é mostrado na Figura 5.39 (b). A fonte aceita dados da camada de transporte e gera um pacote com a rede comum cabeçalho da camada de trabalho, que é IP neste exemplo. O cabeçalho da rede contém o

Página 452

428

A CAMADA DE REDE

INDIVÍDUO. 5

802.11

MPLS

Ethernet

Fonte

Destino

Pacote

Círculo virtual

802.11

IP

IP

Roteador

Roteador

802.11

IP

IP

IP MPLS

Eth

IP

IP

MPLS

IP

IP

Eth IP

Física

(uma)

(b)

Dados de

camada de transporte

Figura 5-39. (a) Um pacote que cruza diferentes redes. (b) Rede e link

processamento de protocolo de camada.

endereço de destino final, que é usado para determinar que o pacote deve ser enviado através do primeiro roteador. Assim, o pacote é encapsulado em um quadro 802.11 cujo destino é o primeiro roteador e transmitido. No roteador, o pacote é removido do campo de dados do quadro e o cabeçalho do quadro 802.11 é descartado. O roteador agora examina o endereço IP no pacote e procura esse endereço em seu roteamento tabela. Com base neste endereço, ele decide enviar o pacote para o segundo roteador Próximo. Para esta parte do caminho, um circuito virtual MPLS deve ser estabelecido para o segundo roteador e o pacote devem ser encapsulados com cabeçalhos MPLS que viajam este circuito. Na extremidade oposta, o cabeçalho MPLS é descartado e o endereço de rede é consultado novamente para encontrar o próximo salto da camada de rede. É o próprio destino. Como o pacote é muito longo para ser enviado pela Ethernet, ele é dividido em duas partes. Cada uma dessas partes é colocada no campo de dados de um quadro Ethernet e enviada para o endereço Ethernet do destino. No destino, o cabeçalho Ethernet é retirado de cada um dos quadros e o conteúdo é remontado. O pacote finalmente alcançou seu destino.

Observe que há uma diferença essencial entre o caso roteado e o caso comutado (ou em ponte). Com um roteador, o pacote é extraído do quadro e o endereço de rede no pacote é usado para decidir para onde enviá-lo. Com um switch (ou bridge), todo o quadro é transportado com base em seu anúncio MAC vestir. Os switches não precisam entender o protocolo da camada de rede que está sendo usado para trocar pacotes. Os roteadores sim.

Infelizmente, a internetworking não é tão fácil quanto parece. No

Na verdade, quando as pontes foram introduzidas, pretendia-se que elas se unissem a diferentes tipos de redes ou, pelo menos, diferentes tipos de LANs. Eles deveriam fazer isso por traduzir frames de uma LAN em frames de outra LAN. No entanto, este

Página 453

SEC. 5,5

INTERNETWORKING

429

não funcionou bem, pela mesma razão que a internetworking é difícil: a diferença nas características das LANs, como diferentes tamanhos máximos de pacotes e LANs com e sem classes de prioridade são difíceis de mascarar. Hoje, as pontes são predominantemente usadas para conectar o mesmo tipo de rede na camada de enlace, e os roteadores conectam diferentes redes na camada de rede.

Internetworking tem tido muito sucesso na construção de grandes redes, mas só funciona quando há uma camada de rede comum. Na verdade, houve muitos protocolos de rede ao longo do tempo. Fazer com que todos concordem em um único formato é difícil quando as empresas percebem que é uma vantagem comercial ter um formato proprietário que eles controlam. Exemplos além do IP, que agora é o protocolo de rede quase universal, foram IPX, SNA e AppleTalk. Nenhum desses protocolos ainda são amplamente usados, mas sempre haverá outros protocolos. O exemplo mais relevante agora é provavelmente IPv4 e IPv6. Embora estes sejam ambos versões de IP, eles não são compatíveis (ou não teria sido necessário criar IPv6).

Um roteador que pode lidar com vários protocolos de rede é chamado de **multiprotocolo roteador**. Deve traduzir os protocolos ou deixar a conexão para um superior camada de protocolo. Nenhuma das abordagens é inteiramente satisfatória. Conexão em um nível superior

camada, digamos, usando TCP, requer que todas as redes implementem TCP (que pode não ser o caso). Então, ele limita o uso nas redes para aplicativos que use TCP (que não inclui muitos aplicativos de tempo real).

A alternativa é traduzir pacotes entre as redes. No entanto, a menos que os formatos de pacote são parentes próximos com os mesmos campos de informação, como as conversões sempre serão incompletas e frequentemente fadadas ao fracasso. Por exemplo, Os endereços IPv6 têm 128 bits. Eles não cabem em um campo de endereço IPv4 de 32 bits, não importa o quanto o roteador tente. Fazer com que IPv4 e IPv6 rodam na mesma rede o trabalho tem se mostrado um grande obstáculo para a implantação do IPv6. (Para ser justo, então faz com que os clientes entendam por que eles deveriam querer o IPv6 em primeiro lugar.) Maiores problemas podem ser esperados ao traduzir entre fundamentalmente diferentes protocolos diferentes, como protocolo de rede orientado a conexão e sem conexão cols. Dadas essas dificuldades, a conversão raramente é tentada. Discutivelmente, até mesmo o IP funcionou tão bem servindo como uma espécie de menor denominação comum nator. Requer pouco das redes em que é executado, mas oferece apenas o melhor esforço serviço como resultado.

5.5.3 Tunelamento

Lidar com o caso geral de fazer a interoperação de duas redes diferentes é extremamente difícil. No entanto, há um caso especial comum que é o homem viável mesmo para diferentes protocolos de rede. Este caso é onde a fonte e hosts de destino estão no mesmo tipo de rede, mas há uma rede diferente entre. Por exemplo, pense em um banco internacional com uma rede IPv6

Página 454

430

A CAMADA DE REDE
INDIVÍDUO. 5

em Paris, uma rede IPv6 em Londres e conectividade entre os escritórios via Internet IPv4. Essa situação é mostrada na Figura 5-40.

IPv6
IPv4
IPv6
Paris
Londres
Túnel
Roteador
Roteador
Pacote IPv4 IPv6
Pacote IPv6
Pacote IPv6

Figura 5-40. Enviando um pacote de Paris a Londres.

A solução para esse problema é uma técnica chamada **tunelamento**. Para enviar um IP

pacote para um anfitrião no escritório de Londres, um anfitrião no escritório de Paris constrói o pacote contendo um endereço IPv6 em Londres, e o envia para o multiprotocolo roteador que conecta a rede IPv6 de Paris à Internet IPv4. Quando este roteador obtém o pacote IPv6, ele encapsula o pacote com um cabeçalho IPv4 endereçado a o lado IPv4 do roteador multiprotocolo que se conecta à rede IPv6 de Londres trabalhos. Ou seja, o roteador coloca um pacote (IPv6) dentro de um pacote (IPv4). Quando isso pacote embrulhado chega, o roteador de Londres remove o pacote IPv6 original e o envia para o host de destino.

O caminho através da Internet IPv4 pode ser visto como um grande túnel que se estende desde um roteador multiprotocolo para o outro. O pacote IPv6 apenas viaja de uma extremidade do túnel para o outro, confortável em sua bela caixa. Não precisa se preocupar com lidar com IPv4 em tudo. Nem os anfitriões em Paris ou Londres. Apenas o multi os roteadores de protocolo precisam entender os pacotes IPv4 e IPv6. Na verdade, o en- A viagem de pneu de um roteador multiprotocolo ao outro é como saltar sobre um único link. Uma analogia pode tornar o tunelamento mais claro. Considere uma pessoa dirigindo seu carro de Paris a Londres. Na França, o carro se move por conta própria, mas quando atinge o Canal da Mancha, é carregado em um trem de alta velocidade e tran- transportado para a Inglaterra através do Chunnel (carros não estão autorizados a passar o Chunnel). Efetivamente, o carro está sendo transportado como carga, conforme descrito na Fig. 5-41. Na extremidade oposta, o carro é solto nas estradas inglesas e mais uma vez continua a se mover por conta própria. Tunelamento de pacotes através de um estrangeiro rede funciona da mesma maneira.

O encapsulamento é amplamente usado para conectar hosts e redes isolados usando outros redes. A rede resultante é chamada de **sobreposição**, uma vez que efetivamente foi sobreposto na rede de base. Implantação de um protocolo de rede com um novo recurso é um motivo comum, como mostra nosso exemplo de " IPv6 sobre IPv4 ". A desgraça vantagem do tunelamento é que nenhum dos hosts na rede que é tunelado pode ser alcançado porque os pacotes não podem escapar no meio do túnel.

Página 455

SEC. 5,5
INTERNETWORKING

431

Carro
canal inglês
Paris
Londres
Ferrovia
Carruagem

Figura 5-41. Túnel de um carro da França para a Inglaterra.

No entanto, essa limitação de túneis é transformada em uma vantagem com **VPNs (Redes privadas virtuais)**. Uma VPN é simplesmente uma sobreposição usada para fornecer um medida de segurança. Exploraremos VPNs quando chegarmos ao cap. 8

5.5.4 Roteamento Internetwork

O roteamento através de uma Internet apresenta o mesmo problema básico que o roteamento dentro de um rede única, mas com algumas complicações adicionais. Para começar, as redes podem usar internamente diferentes algoritmos de roteamento. Por exemplo, uma rede pode usar roteamento de estado de link e outro roteamento de vetor de distância. Uma vez que algoritmos de estado de link

precisa saber a topologia, mas os algoritmos do vetor de distância não, essa diferença por si só, não fica claro como encontrar os caminhos mais curtos na Internet.

Redes operadas por operadoras diferentes levam a problemas maiores. Primeiro, os roteadores podem ter idéias diferentes sobre o que é um bom caminho através da rede. Um operador pode querer a rota com o menor atraso, enquanto outro pode querer rota mais barata. Isso levará os operadores a usar diferentes quantidades para definir os custos do caminho mais curto (por exemplo, milissegundos de atraso vs. custo monetário).

pesos não serão comparáveis entre redes, então os caminhos mais curtos na internet

não será bem definido.

Pior ainda, um operador pode não querer outro operador nem mesmo saber o de-caudas dos caminhos em sua rede, talvez porque os pesos e caminhos podem refletir informações confidenciais (como o custo monetário) que representam uma vantagem comercial.

Finalmente, a internet pode ser muito maior do que qualquer uma das redes que compreendê-lo. Pode, portanto, exigir algoritmos de roteamento que escalam bem usando um hierarquia, mesmo que nenhuma das redes individuais precise usar uma hierarquia.

Todas essas considerações levam a um algoritmo de roteamento de dois níveis. Dentro de cada rede, um **protocolo intradomínio** ou de **gateway interior** é usado para o roteamento.

("Gateway" é um termo mais antigo para "roteador"). Pode ser um protocolo de estado de link do tipo que já descrevemos. Nas redes que constituem a Internet, um **protocolo de gateway externo** ou **interdomínio** é usado. Todas as redes podem usar diferentes protocolos intradomínio, mas devem usar o mesmo protocolo interdomínio.

Página 456

432

A CAMADA DE REDE INDIVÍDUO. 5

Na Internet, o protocolo de roteamento entre domínios é denominado **BGP (Border Gateway Protocol)**. Nós o descreveremos na próxima seção.

Há mais um termo importante a ser introduzido. Uma vez que cada rede está operando independentemente de todos os outros, é muitas vezes referido como **AS (Autônomo System)**. Um bom modelo mental para um AS é uma rede ISP. Na verdade, uma rede ISP o trabalho pode ser composto por mais de um AS, se for gerenciado, ou, foi ac- exigido, como várias redes. Mas a diferença geralmente não é significativa.

Os dois níveis geralmente não são estritamente hierárquicos, como caminhos altamente subótimos poderia resultar se uma grande rede internacional e uma pequena rede regional fossem ambos abstraídos para ser uma única rede. No entanto, relativamente pouca informação sobre as rotas dentro das redes é exposta para encontrar rotas através da internetwork.

Isso ajuda a resolver todas as complicações. Melhora o dimensionamento e permite o funcionamento tors selecionam livremente rotas dentro de suas próprias redes usando um protocolo de sua escolha ing. Também não requer que os pesos sejam comparados entre redes ou exponha informações sensíveis fora das redes.

No entanto, falamos pouco até agora sobre como as rotas através das redes da Internet são determinados. Na Internet, um grande fator determinante é o acordos comerciais entre ISPs. Cada ISP pode cobrar ou receber dinheiro de outros ISPs para transportar tráfego. Outro fator é que se internetwork roteamento exige cruzar fronteiras internacionais, várias leis podem repentinamente entram em jogo, como as rígidas leis de privacidade da Suécia sobre exportação dados sobre cidadãos suecos da Suécia. Todos esses fatores não técnicos são embrulhado no conceito de uma **política de roteamento** que rege a forma autônoma redes selecionam as rotas que usam. Voltaremos às políticas de roteamento quando nós descrevemos o BGP.

5.5.5 Fragmentação de Pacotes

Cada rede ou link impõe algum tamanho máximo em seus pacotes. Estes lim- tem várias causas, entre elas

1. Hardware (por exemplo, o tamanho de um quadro Ethernet).
2. Sistema operacional (por exemplo, todos os buffers têm 512 bytes).
3. Protocolos (por exemplo, o número de bits no campo de comprimento do pacote).
4. Conformidade com algum padrão (inter) nacional.
5. Desejo de reduzir as retransmissões induzidas por erro a algum nível.
6. Desejo de evitar que um pacote ocupe o canal por muito tempo.

O resultado de todos esses fatores é que os designers da rede não são livres para escolher qualquer tamanho de pacote máximo antigo que desejarem. Cargas úteis máximas para alguns

SEC. 5,5
INTERNETWORKING

433

tecnologias são 1500 bytes para Ethernet e 2272 bytes para 802.11. IP é mais generoso, permite pacotes de até 65.515 bytes.

Os hosts geralmente preferem transmitir pacotes grandes porque isso reduz o pacote sobrecargas, como largura de banda desperdiçada em bytes de cabeçalho. Um óbvio internetwork-problema surge quando um grande pacote quer viajar através de uma rede cujo o tamanho máximo do pacote é muito pequeno. Esse incômodo tem sido um problema persistente e as soluções para isso evoluíram junto com a experiência adquirida na Internet.

Uma solução é garantir que o problema não ocorra em primeiro lugar.

No entanto, isso é mais fácil dizer do que fazer. Uma fonte geralmente não conhece o caminho de um

pacote vai levar através da rede para um destino, então certamente não saber quão pequenos os pacotes devem ser para chegar lá. Este tamanho de pacote é chamado de **Caminho**

MTU (unidade de transmissão máxima do caminho). Mesmo se a fonte conhecesse o caminho MTU, os pacotes são roteados de forma independente em uma rede sem conexão, como a Internet. Este roteamento significa que os caminhos podem mudar repentinamente, o que pode mudar inesperadamente o caminho MTU.

A solução alternativa para o problema é permitir que os roteadores dividam os pacotes em **fragmentos**, enviando cada fragmento como um pacote de camada de rede separado. Como sempre, como todo pai de uma criança pequena sabe, converter um objeto grande em pequeno fragmentos é consideravelmente mais fácil do que o processo reverso. (Os físicos têm até deu a este efeito um nome: a segunda lei da termodinâmica.) Comutação de pacotes as redes também têm problemas para reunir os fragmentos novamente.

Existem duas estratégias opostas para recombinar os fragmentos de volta ao pacote original. A primeira estratégia é fazer a fragmentação causada por um " pequeno pacote " rede transparente para quaisquer redes subsequentes através das quais o pacote deve passar em seu caminho para o destino final. Esta opção é mostrada na Fig. 5-42 (a). Nesta abordagem, quando um pacote superdimensionado chega em G_1 , o roteador quebra em fragmentos. Cada fragmento é endereçado ao mesmo roteador de saída, G_2 , onde as peças são recombinadas. Desta forma, a passagem pelo pacote pequeno rede torna-se transparente. As redes subsequentes nem mesmo estão cientes de que fragmentação ocorreu.

A fragmentação transparente é direta, mas tem alguns problemas. Para um coisa, o roteador de saída deve saber quando recebeu todas as peças, então um campo de contagem ou um bit de " fim de pacote " deve ser fornecido. Além disso, porque todos os pacotes

devem sair através do mesmo roteador para que possam ser remontados, as rotas são constreñidas. Por não permitir que alguns fragmentos sigam uma rota para o destino final inação e outros fragmentos uma rota disjunta, algum desempenho pode ser perdido. Mais significativo é a quantidade de trabalho que o roteador pode ter que realizar. Pode ser necessário armazene os fragmentos conforme eles chegam e decida quando jogá-los fora, se não todos dos fragmentos chegam. Parte desse trabalho também pode ser um desperdício, pois o pacote pode passar por uma série de redes de pequenos pacotes e precisa ser repetidamente fragmentado e remontado.

A outra estratégia de fragmentação é abster-se de recombinar fragmentos em quaisquer roteadores intermediários. Uma vez que um pacote foi fragmentado, cada fragmento é

434

A CAMADA DE REDE
INDIVÍDUO. 5

G_1
 G_2
 G_3

```

G 4
G 1
G 2
G 3
G 4
Pacote
Rede 1
Fragmentos G 1
um pacote grande
G 2
remonta
os fragmentos
Fragmentos G 3
novamente
G 4
remonta
novamente
Rede 2
(uma)
Pacote
Fragmentos G 1
um pacote grande
Os fragmentos não são remontados
até que o destino final (um host) seja alcançado
(b)

```

Figura 5-42. (a) Fragmentação transparente. (b) Fragmentação não transparente.
tratado como se fosse um pacote original. Os roteadores passam os fragmentos, como mostrado na Figura 5-42 (b), e a remontagem é realizada apenas no host de destino. A principal vantagem da fragmentação não transparente é que ela requer roteadores para fazer menos trabalho. O IP funciona dessa maneira. Um design completo requer que os fragmentos ser numerados de forma que o fluxo de dados original possa ser reconstruído. O design usado pelo IP é dar a cada fragmento um número de pacote (transportado em todos pacotes), um deslocamento de byte absoluto dentro do pacote e uma bandeira indicando se é o fim do pacote. Um exemplo é mostrado na Figura 5-43. Embora simples, este design tem algumas propriedades atraentes. Os fragmentos podem ser colocados em um buffer no destino no local certo para remontagem, mesmo que cheguem fora de serviço. Fragmentos também podem ser fragmentados se passarem por uma rede com um ainda menor MTU. Isso é mostrado na Figura 5-43 (c). Retransmissões do pacote (se todos fragmentos não foram recebidos) podem ser fragmentados em partes diferentes. Finalmente, fragmentos podem ser de tamanho arbitrário, até um único byte mais o cabeçalho do pacote. Em todos os casos, o destino simplesmente usa o número do pacote e deslocamento do fragmento para colocar os dados na posição certa, e o sinalizador de fim do pacote para determinar quando ele o pacote completo. Infelizmente, esse design ainda apresenta problemas. A sobrecarga pode ser maior do que com a fragmentação transparente porque os cabeçalhos dos fragmentos agora são carregados sobre alguns links onde eles podem não ser necessários. Mas o verdadeiro problema é a existência da presença de fragmentos em primeiro lugar. Kent e Mogul (1987) argumentaram que fragmentação é prejudicial ao desempenho porque, assim como as sobrecargas do cabeçalho, um pacote inteiro é perdido se algum de seus fragmentos for perdido, e porque a fragmentação é mais um fardo para os hosts do que se pensava originalmente.

```

Eu
J
27
0 0 A
B
C
D
E
F
G
H
27
8 1
Eu
J
27
0 0 A
B
C
D
E
27
5 0 F
G
H
27
8 1
Eu
J
Cabeçalho
1 byte
Cabeçalho
Cabeçalho
Cabeçalho
Cabeçalho
Cabeçalho
Cabeçalho
(uma)
(b)
(c)

```

Figura 5-43. Fragmentação quando o tamanho dos dados elementares é de 1 byte. (a) Orig-pacote final, contendo 10 bytes de dados. (b) Fragmentos depois de passar por uma rede trabalhar com tamanho máximo de pacote de 8 bytes de carga útil mais o cabeçalho. (c) Fragmentos depois de passar por um gateway tamanho 5.

Isso nos leva de volta à solução original de se livrar da fragmentação em a rede, a estratégia usada na Internet moderna. O processo é chamado de **caminho**

Descoberta MTU (Mogul e Deering, 1990). Funciona da seguinte maneira. Cada pacote IP é enviado com seus bits de cabeçalho definidos para indicar que nenhuma fragmentação pode ser realizada. Se um roteador receber um pacote muito grande, ele gerará um erro pacote, retorna-o à origem e descarta o pacote. Isso é mostrado na Figura 5-44.

Quando a fonte recebe o pacote de erro, ela usa as informações dentro para refazer

Mente o pacote em pedaços pequenos o suficiente para serem manuseados pelo roteador. Se um roteador mais adiante no caminho tem um MTU ainda menor, o processo é repetido.

Fonte	
Destino	
Pacote (com comprimento)	
“Tente 900”	
“Experimente 1200”	
1200	
900	
1400	

Figura 5-44. Descoberta de MTU de caminho.

A vantagem da descoberta de caminho MTU é que a fonte agora sabe o que pacote de comprimento para enviar. Se as rotas e o caminho MTU mudarem, novos pacotes de erro irão

ser acionado e a fonte se adaptará ao novo caminho. No entanto, a fragmentação é ainda é necessário entre a origem e o destino, a menos que as camadas superiores aprendam o caminho MTU e passar a quantidade certa de dados para o IP. TCP e IP são normalmente implementados juntos (como "TCP / IP") para poder passar este tipo de informação.

Mesmo que isso não seja feito para outros protocolos, a fragmentação ainda foi removida

da rede e nos hosts.

A desvantagem da descoberta do caminho MTU é que pode haver inicialização adicional atrasos simplesmente para enviar um pacote. Mais de um atraso de ida e volta pode ser necessário para

teste o caminho e encontre a MTU antes que qualquer dado seja entregue ao destino.

Isso levanta a questão de saber se existem projetos melhores. A resposta é provavelmente "Sim". Considere o projeto em que cada roteador simplesmente trunca os pacotes que exceder seu MTU. Isso garantiria que o destino aprenda o MTU tão rapidamente quanto possível (a partir da quantidade de dados que foi entregue) e recebe alguns dos dados.

5.6 A CAMADA DE REDE NA INTERNET

Agora é hora de discutir a camada de rede da Internet em detalhes. Mas depois entrando em detalhes, vale a pena dar uma olhada nos princípios que impulsionaram sua decadência assinar no passado e torná-lo o sucesso que é hoje. Com muita frequência, hoje em dia, as pessoas parecem tê-los esquecido. Esses princípios são enumerados e discutidos na RFC 1958, que vale a pena ler (e deve ser obrigatório para todos designers de protocolo - com um exame final no final). Este RFC baseia-se fortemente em ideias apresentadas por Clark (1988) e Saltzer et al. (1984). Vamos agora resumir o que consideramos ser os 10 principais princípios (do mais importante para o menos importante tanto).

1. Certifique-se de que funciona. Não finalize o design ou padrão até vários protótipos se comunicarem com sucesso com cada um de outros. Muito frequentemente, os designers primeiro escrevem um padrão de 1000 páginas, entendam aprovado, então descubra que é profundamente falho e não funciona. Então eles escrevem a versão 1.1 do padrão. Este não é o caminho a percorrer.

2. Mantenha a simplicidade. Em caso de dúvida, use a solução mais simples. William de Occam declarou este princípio (a navalha de Occam) no século XIV. Colocado em termos modernos: recursos de luta. Se um recurso não for absolutamente essencial, deixe-o de fora, especialmente se o mesmo efeito pode ser alcançado por combinando outros recursos.

3. Faça escolhas claras. Se houver várias maneiras de fazer o mesmo coisa, escolha um. Ter duas ou mais maneiras de fazer a mesma coisa é procurando problemas. Os padrões geralmente têm várias opções ou modos

Página 461

SEC. 5,6

A CAMADA DE REDE NA INTERNET

437

ou parâmetros, porque vários partidos poderosos insistem que o seu caminho é melhor. Os designers devem resistir fortemente a essa tendência. Apenas diga não.

4. Explorar a modularidade. Este princípio leva diretamente à ideia de ter em pilhas de protocolo, cada uma das camadas é independente de todos os outros. Desta forma, se as circunstâncias exigirem um módulo ou camada a serem alterados, os outros não serão afetados.

5. Espere heterogeneidade. Diferentes tipos de hardware, transmissão instalações e aplicativos ocorrerão em qualquer grande rede. Para lidar com eles, o design da rede deve ser simples, geral e flexível.

6. Evite opções e parâmetros estáticos. Se os parâmetros forem inevitáveis-capaz (por exemplo, tamanho máximo do pacote), é melhor ter o remetente e voltar ceiver negociar um valor em vez de definir escolhas fixas.

7. Procure um bom design; não precisa ser perfeito. Freqüentemente, os signatários têm um bom design, mas não podem lidar com alguns especiais estranhos caso. Em vez de bagunçar o design, os designers deveriam ir com o bom design e colocar o fardo de trabalhar em torno disso no pessoas com os requisitos estranhos.

- 8. Seja rigoroso ao enviar e tolerante ao receber.** Em outro palavras, envie apenas pacotes que cumpram rigorosamente os padrões, mas espere pacotes de entrada que podem não estar totalmente em conformidade e tente lidar com eles.
- 9. Pense em escalabilidade.** Se o sistema deve lidar com milhões de hosts e bilhões de usuários efetivamente, nenhum banco de dados centralizado de qualquer tipo são toleráveis e a carga deve ser distribuída o mais uniformemente possível sobre os recursos disponíveis.
- 10. Considere o desempenho e o custo.** Se uma rede tem desempenho ruim custo excessivo ou exorbitantes, ninguém o usará.
- Vamos agora deixar os princípios gerais e começar a olhar para os detalhes da Camada de rede da Internet. Na camada de rede, a Internet pode ser vista como uma coleção de redes ou **ASes (Sistemas Autônomos)** que estão interconectados. Não existe uma estrutura real, mas existem vários backbones principais. Estes são construídos a partir de linhas de alta largura de banda e roteadores rápidos. A maior dessas costas bones, aos quais todos se conectam para acessar o resto da Internet, são chamados **Redes de nível 1**. Ligados aos backbones estão os ISPs (Internet Service Providers) que fornecem acesso à Internet para residências e empresas, data centers e instalações de colocation cheias de máquinas de servidor e redes regionais (nível médio). Os data centers atendem a grande parte do conteúdo enviado pela Internet. Em anexo

Página 462

438

A CAMADA DE REDE INDIVÍDUO. 5

para as redes regionais são mais ISPs, LANs em muitas universidades e com empresas e outras redes de ponta. Um esboço desta organização quase hierárquica é apresentado na Figura 5-45.

Linhos alugados
para a ásia
Um backbone dos EUA
Alugado
transatlântico
linhos
Uma espinha dorsal europeia
Nacional
rede
Companhia
rede
Ethernet
Roteador IP
Móvel
rede
WiMAX
Cabo
Casa
rede
Regional
rede

Figura 5-45. A Internet é uma coleção interconectada de muitas redes.

A cola que mantém toda a Internet unida é o protocolo da camada de rede, **IP (protocolo da Internet)**. Ao contrário da maioria dos protocolos da camada de rede mais antigos, o IP era desenhado desde o início com a internetworking em mente. Uma boa maneira de pensar a camada de rede é esta: seu trabalho é fornecer o melhor esforço (ou seja, não garantido) maneira de transportar pacotes da origem ao destino, independentemente de essas máquinas estão na mesma rede ou se existem outras redes entre elas.

A comunicação na Internet funciona da seguinte maneira. A camada de transporte leva fluxos de dados e os divide para que possam ser enviados como pacotes IP. Em teoria, os pacotes podem ter até 64 KB cada, mas na prática eles geralmente não são maiores do que 1500 bytes (para que se encaixem em um quadro Ethernet). Os roteadores IP encaminham cada pacote pela Internet, ao longo de um caminho de um roteador a outro, até o destino.

é atingido. No destino, a camada de rede entrega os dados para o transporte camada, que o entrega ao processo de recebimento. Quando todas as peças finalmente chegarem a máquina de destino, eles são remontados pela camada de rede na origem datagrama final. Este datagrama é então entregue à camada de transporte.

No exemplo da Figura 5.45, um pacote originado em um host na rede doméstica trabalho tem que atravessar quatro redes e um grande número de roteadores IP antes mesmo chegar à rede da empresa na qual o host de destino está localizado. Isto é

Página 463

SEC. 5,6

A CAMADA DE REDE NA INTERNET

439

não incomum na prática, e existem muitos caminhos mais longos. Também tem muito conectividade redundante na Internet, com backbones e ISPs conectando-se a uns aos outros em vários locais. Isso significa que existem muitos caminhos possíveis entre dois hosts. É função dos protocolos de roteamento IP decidir quais caminhos usar.

5.6.1 O Protocolo IP Versão 4

Um lugar apropriado para começar nosso estudo da camada de rede na Internet é com o formato dos próprios datagramas IP. Um datagrama IPv4 consiste em um parte do cabeçalho e um corpo ou parte da carga útil. O cabeçalho tem uma parte fixa de 20 bytes e um

parte opcional de comprimento variável. O formato do cabeçalho é mostrado na Figura 5-46. Os bits são transmitidos da esquerda para a direita e de cima para baixo, com o bit de ordem superior do *O campo da versão* vai primeiro. (Esta é uma ordem de bytes de rede "big-endian". máquinas endian, como computadores Intel x86, uma conversão de software é necessária tanto na transmissão quanto na recepção.) Em retrospecto, little endian teria sido uma escolha melhor, mas na época em que o IP foi projetado, ninguém sabia que chegaria a dominar a computação.

Versão
IHL
Comprimento total
Tempo de Viver
Protocolo
Serviços diferenciados
Identificação
Soma de verificação do cabeçalho
Deslocamento de fragmento
Endereço de Origem
Endereço de destino
Opções (0 ou mais palavras)
D
F
M
F
32 bits

Figura 5-46. O cabeçalho IPv4 (Internet Protocol).

O campo *Versão* mantém registro de qual versão do protocolo o datagrama pertence a. A versão 4 domina a Internet hoje, e é aí que temos começado nossa discussão. Ao incluir a versão no início de cada datagrama, torna-se possível ter uma transição entre as versões durante um longo período de tempo. Na verdade, o IPv6, a próxima versão do IP, foi definido há mais de uma década, mas ainda é apenas começando a ser implantado. Iremos descrevê-lo mais tarde nesta seção. Está o uso acabará sendo forçado quando cada uma das quase 2³¹ pessoas da China tiver uma mesa PC superior, um laptop e um telefone IP. Como um aparte na numeração, IPv5 foi uma experiência protocolo de fluxo em tempo real imental que nunca foi amplamente usado.

Página 464

440

A CAMADA DE REDE
INDIVÍDUO. 5

Uma vez que o comprimento do cabeçalho não é constante, um campo no cabeçalho, *IHL*, é fornecido

para saber o comprimento do cabeçalho, em palavras de 32 bits. O valor mínimo é 5, que aplica-se quando nenhuma opção está presente. O valor máximo deste campo de 4 bits é 15, que limita o cabeçalho a 60 bytes e, portanto, o campo *Opções* a 40 bytes. Para algumas opções, como aquela que registra a rota percorrida por um pacote, 40 bytes está longe muito pequeno, tornando essas opções inúteis.

O campo de *serviços diferenciados* é um dos poucos campos que mudou seu significado (ligeiramente) ao longo dos anos. Originalmente, era chamado de *tipo de serviço* campo. Foi e ainda se destina a distinguir entre diferentes classes de serviço. Várias combinações de confiabilidade e velocidade são possíveis. Para digitalizado voz, entrega rápida bate entrega precisa. Para transferência de arquivos, transmissão sem erros A transmissão é mais importante do que a transmissão rápida. O campo *Tipo de serviço* fornecido 3 bits para sinalizar a prioridade e 3 bits para sinalizar se um host se preocupa mais com o atraso, rendimento ou confiabilidade. No entanto, ninguém sabia realmente o que fazer com esses bits em roteadores, por isso não foram usados por muitos anos. Quando serviços diferenciados foram projetados, o IETF jogou a toalha e reutilizou este campo. Agora, os 6 principais bits são usados para marcar o pacote com sua classe de serviço; descrevemos o acelerado e serviços garantidos anteriormente neste capítulo. Os 2 bits inferiores são usados para transportar informações de notificação de congestionamento cit, como se o pacote tem experiência congestionamento enced; descrevemos a notificação explícita de congestionamento como parte dos congestionamentos controle de instalação anteriormente neste capítulo.

O *comprimento total* inclui tudo no datagrama - cabeçalho e dados.

O comprimento máximo é de 65.535 bytes. No momento, esse limite superior é tolerável, mas com redes futuras, datagramas maiores podem ser necessários.

O campo *Identificação* é necessário para permitir que o host de destino determine a qual pacote um fragmento recém-chegado pertence. Todos os fragmentos de um pacote contêm o mesmo valor de *identificação*.

Em seguida, vem uma parte não utilizada, o que é surpreendente, já que os imóveis disponíveis no O cabeçalho IP é extremamente escasso. Como uma piada do primeiro de abril, Bellovin (2003) propôs

usando este bit para detectar tráfego malicioso. Isso simplificaria muito a segurança, pois pacotes com o conjunto de bits " mal " seriam conhecidos por terem sido enviados por invasores e poderia simplesmente ser descartado. Infelizmente, a segurança da rede não é tão simples.

Em seguida, vêm dois campos de 1 bit relacionados à fragmentação. *DF* significa não Fragmento. É uma ordem para os roteadores não fragmentarem o pacote. Originalmente, foi projetado para apoiar hosts incapazes de juntar as peças novamente.

Agora ele é usado como parte do processo para descobrir o caminho MTU, que é o maior pacote est que pode viajar ao longo de um caminho sem ser fragmentado. Marcando o datagrama com o bit *DF*, o remetente sabe que chegará inteiro ou um a mensagem de erro será devolvida ao remetente.

MF significa More Fragments. Todos os fragmentos, exceto o último, têm este conjunto de bits. É necessário saber quando todos os fragmentos de um datagrama chegaram.

O *deslocamento do fragmento* informa a que parte do pacote atual esse fragmento pertence. Todos os fragmentos, exceto o último em um datagrama, devem ser múltiplos de 8 bytes, o

unidade de fragmento elementar. Uma vez que 13 bits são fornecidos, há um máximo de 8192 fragmentos por datagrama, suportando um comprimento máximo de pacote até o limite de o campo *Comprimento total*. Trabalhando juntos, a *Identificação*, o *MF* e o *Fragmento* campos de *deslocamento* são usados para implementar a fragmentação conforme descrito na Seç. 5.5.5.

O campo *TtL* (*Time to live*) é um contador usado para limitar a vida útil dos pacotes. isso foi

originalmente deveria contar o tempo em segundos, permitindo uma vida útil máxima de 255 seg. Deve ser diminuído em cada salto e deve ser diminuído simultaneamente várias vezes quando um pacote é enfileirado por um longo tempo em um roteador. Na prática, é apenas conta lúpulos. Quando chega a zero, o pacote é descartado e um pacote de aviso é enviado de volta ao host de origem. Este recurso evita que os pacotes vaguem para sempre, algo que poderia acontecer se as tabelas de roteamento sempre se corromper.

Quando a camada de rede montou um pacote completo, ela precisa saber O que fazer com isso. O campo *Protocolo* informa qual processo de transporte dar o pacote para. O TCP é uma possibilidade, mas o UDP e alguns outros também. O número A combinação de protocolos é global em toda a Internet. Protocolos e outras atribuições números ed foram anteriormente listados no RFC 1700, mas hoje em dia eles estão contidos em um banco de dados online localizado em www.iana.org. Uma vez que o cabeçalho carrega informações vitais, como endereços, ele classifica seus próprios checksum para proteção, o *checksum do cabeçalho*. O algoritmo é somar todos os Halfwords de 16 bits do cabeçalho à medida que chegam, usando a aritmética do complemento de alguém, e então pegue o complemento de um do resultado. Para fins deste algoritmo, a *soma de verificação do cabeçalho* é considerada zero na chegada. Essa soma de verificação é útil para detectar erros enquanto o pacote viaja pela rede. Observe que deve ser recalculado em cada salto porque pelo menos um campo sempre muda (o *Time to live field*), mas truques podem ser usados para acelerar o cálculo. O *endereço de origem* e o *endereço de destino* indicam o endereço IP do interfaces de rede de origem e destino. Discutiremos endereços de Internet em a próxima seção.

O campo *Opções* foi projetado para fornecer um escape para permitir versões subsequentes seções do protocolo para incluir informações não presentes no projeto original, para permitem que os experimentadores experimentem novas idéias e evitem alocar bits de cabeçalho para informações que raramente são necessárias. As opções são de tamanho variável. Cada um começa com um código de 1 byte identificando a opção. Algumas opções são seguidas por um 1 byte campo de comprimento de opção e, em seguida, um ou mais bytes de dados. O campo *Opções* é preenchido para um múltiplo de 4 bytes. Originalmente, as cinco opções listadas na Figura 5-47 eram definiram.

A opção *Segurança* informa o quanto secreta é a informação. Em teoria, um militar roteador pode usar este campo para especificar não rotear pacotes através de certos países os militares consideram "bandidos". Na prática, todos os roteadores o ignoram, então é a única função prática é ajudar os espiões a encontrarem mais facilmente as coisas boas. A opção de *roteamento de origem estrita* fornece o caminho completo da origem ao destino nação como uma sequência de endereços IP. O datagrama é necessário para seguir esse

Figura 5-47. Algumas das opções de IP.

rota exata. É mais útil para gerentes de sistema que precisam enviar emergência pacotes quando as tabelas de roteamento foram corrompidas, ou para fazer medições de tempo.

A opção de *roteamento de origem frouxa* requer que o pacote atravesse a lista de roteadores especificados, na ordem especificada, mas é permitido passar por outros roteadores no caminho. Normalmente, esta opção fornecerá apenas alguns roteadores, para forçar um caminho particular. Por exemplo, para forçar um pacote de Londres a Sydney a ir oeste em vez de leste, esta opção pode especificar roteadores em Nova York, Los Angeles, e Honolulu. Esta opção é mais útil quando se considera política ou econômica razões ditam a passagem ou evitação de certos países.

A opção *Record route* diz a cada roteador ao longo do caminho para anexar seu anúncio IP vestido para o campo *Opções*. Isso permite que os gerentes de sistema rastreiem bugs nos algoritmos de roteamento ("Por que os pacotes de Houston para Dallas estão visitando Tóquio primeiro?"). Quando a ARPANET foi configurada pela primeira vez, nenhum pacote passou por mais de nove roteadores, portanto, 40 bytes de opções eram suficientes. Como acima mencionado, agora é muito pequeno.

Finalmente, a opção *Timestamp* é como a opção *Record route*, exceto que em além de registrar seu endereço IP de 32 bits, cada roteador também registra um tempo de 32 bits carimbo. Essa opção também é útil principalmente para medição de rede.

Hoje, as opções de IP caíram em desuso. Muitos roteadores as ignoram ou fazem não processá-las de forma eficiente, desviando-as para o lado como um caso incomum. que ou seja, eles são apenas parcialmente suportados e raramente são usados.

5.6.2 Endereços IP

Uma característica definidora do IPv4 são seus endereços de 32 bits. Cada host e roteador ativado a Internet tem um endereço IP que pode ser usado no *endereço de origem e destino* campos de *endereço de conexão* de pacotes IP. É importante notar que um endereço IP não se refere realmente a um host. Na verdade, se refere a uma interface de rede, portanto, se um host estiver duas redes, deve ter dois endereços IP. No entanto, na prática, a maioria dos hosts estão em uma rede e, portanto, têm um endereço IP. Em contraste, os roteadores têm vários interfaces e, portanto, vários endereços IP.

Prefixos

Os endereços IP são hierárquicos, ao contrário dos endereços Ethernet. Cada endereço de 32 bits é composta por uma parte da rede de comprimento variável nos bits superiores e uma parte do host nos bits inferiores. A parte da rede tem o mesmo valor para todos os hosts em um único bloco contíguo de espaço de endereço IP. Este bloco é denominado **prefixo**.

Os endereços IP são escritos em **notação decimal com pontos**. Neste formato, cada um dos 4 bytes são escritos em decimais, de 0 a 255. Por exemplo, o hexadecimal de 32 bits

O endereço cimal 80D00297 é escrito como 128.208.2.151. Prefixos são escritos por giving o endereço IP mais baixo do bloco e o tamanho do bloco. O tamanho é determinado

extraído pelo número de bits na parte da rede; os bits restantes no host

porção pode variar. Isso significa que o tamanho deve ser uma potência de dois. Por convenção, ele é escrito após o prefixo do endereço IP como uma barra seguida pelo comprimento em bits da parte da rede. Em nosso exemplo, se o prefixo contém 28 bits, é escrito como 128.208.0.0/24.

Como o comprimento do prefixo não pode ser inferido apenas do endereço IP, o roteamento os protocolos devem levar os prefixos aos roteadores. Às vezes, os prefixos são simplesmente descartados por seu comprimento, como em um "/16" que é pronunciado como "barra 16." O comprimento

do prefixo corresponde a uma máscara binária de 1s na parte da rede. Quando

escrito dessa forma, é chamado de **máscara de sub-rede**. Pode ser AND com o IP ad- vestido para extrair apenas a parte da rede. Para nosso exemplo, a máscara de sub-rede é 255.255.255.0. A Fig. 5-48 mostra um prefixo e uma máscara de sub-rede.

Figura 5-48. Um prefixo IP e uma máscara de sub-rede.

Endereços hierárquicos têm vantagens e desvantagens significativas. A principal vantagem dos prefixos é que os roteadores podem encaminhar pacotes com base apenas no

parte da rede do endereço, desde que cada uma das redes tenha um anúncio exclusivo bloco de vestido. A parte do host não importa para os roteadores porque todos os hosts em a mesma rede será enviada na mesma direção. É só quando os pacotes chegar à rede para a qual se destinam que são encaminhados para o cor-host ret. Isso torna as tabelas de roteamento muito menores do que seriam de outra forma estar. Considere que o número de hosts na Internet está se aproximando de um bilhão. Essa seria uma mesa muito grande para cada roteador manter. No entanto, usando um hierarquia, os roteadores precisam manter as rotas para apenas cerca de 300.000 prefixos.

Página 468

444

A CAMADA DE REDE INDIVÍDUO. 5

Embora o uso de uma hierarquia permita escalar o roteamento da Internet, há duas desvantagens. Primeiro, o endereço IP de um host depende de onde ele está localizado na rede. A O endereço Ethernet pode ser usado em qualquer lugar do mundo, mas cada endereço IP pertence a uma rede específica, e os roteadores só serão capazes de entregar pacotes destinados a esse endereço para a rede. Projetos como IP móvel são necessários para oferecer suporte a hosts que se movem entre redes, mas desejam manter os mesmos endereços IP.

A segunda desvantagem é que a hierarquia desperdiça endereços, a menos que é gerenciado com cuidado. Se os endereços forem atribuídos a redes em (muito) grandes blocos, haverá (muitos) endereços alocados, mas não em uso. Este al-
a localização não importaria muito se houvesse muitos endereços disponíveis.
No entanto, percebeu-se há mais de duas décadas que o tremendo crescimento
da Internet estava esgotando rapidamente o espaço de endereço livre. IPv6 é a solução para
esta escassez, mas até que seja amplamente implantado haverá grande pressão para alocar
Endereços IP para que sejam usados de forma muito eficiente.

Sub-redes

Sab. Tech

Os números da rede são gerenciados por uma corporação sem fins lucrativos chamada **ICANN** (**Internet Corporation for Assigned Names and Numbers**), para evitar conflitos. Por sua vez, ICANN delegou partes do espaço de endereçamento a vários autoridades, que distribuem endereços IP para ISPs e outras empresas. Isto é o processo pelo qual uma empresa recebe um bloco de endereços IP. No entanto, esse processo é apenas o começo da história, pois a atribuição de endereços IP está em curso à medida que as empresas crescem. Dissemos que o roteamento por prefixo requer todos os hosts em uma rede para ter o mesmo número de rede. Esta propriedade pode causar problemas à medida que as redes crescem. Por exemplo, considere uma universidade que começou com nosso prefixo de exemplo / 16 para uso pelo Departamento de Ciência da Computação para os computadores em sua Ethernet. Um ano depois, o Departamento de Engenharia Elétrica deseja obter na internet. O Art Dept. logo segue o exemplo. Quais endereços IP devem esses departamentos usar? Para conseguir mais bloqueios, é preciso sair da universidade e pode ser caro ou inconveniente. Além disso, o / 16 já alocado tem endereços suficientes para mais de 60 000 hosts. Pode ser destinado a permitir significantes

não posso crescer, mas até que isso aconteça, é um desperdício alocar mais blocos de IP endereços para a mesma universidade. É necessária uma organização diferente. A solução é permitir que o bloco de endereços seja dividido em várias partes para uso interno como várias redes, embora ainda atue como uma única rede para o mundo exterior. Isso é chamado de **sub - rede** e as redes (como Ethernet LANs) que resultam da divisão de uma rede maior são chamadas de **sub-redes**. Como nós mencionado no cap. 1, você deve estar ciente de que este novo uso do termo con flitos com o uso mais antigo de " sub-rede " para significar o conjunto de todos os roteadores e comunicações linhas de conexão em uma rede.

A Figura 5-49 mostra como as sub-redes podem ajudar em nosso exemplo. O single / 16 tem sido dividido em pedaços. Esta divisão não precisa ser uniforme, mas cada peça deve ser

Página 469

SEC. 5,6
A CAMADA DE REDE NA INTERNET

445

alinados de forma que quaisquer bits possam ser usados na parte inferior do host. Neste caso, metade de

o bloco (a / 17) é alocado para o Departamento de Ciência da Computação, um quarto é alocado ao Departamento de Engenharia Elétrica (a / 18), e um oitavo (a / 19) ao Departamento de Arte. O oitavo restante não é alocado. Uma maneira diferente de ver como o bloco foi di-

vided é olhar para os prefixos resultantes quando escritos em notação binária:

Ciência da computação: 10000000 11010000 1 | xxxxxxxx xxxxxxxx

Eng eletrico:

10000000 11010000 00 | xxxxxx xxxxxxxx

Arte:

10000000 11010000 011 | xxxx xxxxxxxx

Aqui, a barra vertical (|) mostra o limite entre o número da sub-rede e o porção hospedeira.

Arte

128.208.0.0/16

(para Internet)

128.208.96.0/19

EE

CS

128.208.128.0/17

128.208.0.0/18

Figura 5-49. Dividir um prefixo IP em redes separadas com sub-redes.

Quando um pacote chega ao roteador principal, como o roteador sabe qual sub-rede para dar? É aqui que entram os detalhes dos nossos prefixos.

seria para cada roteador ter uma tabela com 65.536 entradas informando que vai usar a linha para cada host no campus. Mas isso prejudicaria o principal benefício de escala que obtemos usando uma hierarquia. Em vez disso, os roteadores simplesmente precisam

conhecer as máscaras de sub-rede para as redes no campus.

Quando um pacote chega, o roteador olha o endereço de destino do pacote e verifica a qual sub-rede ele pertence. O roteador pode fazer isso colocando o AND no endereço de destino com a máscara para cada sub-rede e verificando se o resultado é o prefixo correspondente. Por exemplo, considere um pacote destinado a um anúncio de IP vestido 128.208.2.151. Para ver se é para o Departamento de Ciência da Computação, nós E com 255.255.128.0 para pegar os primeiros 17 bits (que é 128.208.0.0) e ver se eles correspondem ao endereço de prefixo (que é 128.208.128.0). Eles não combinam. Verificando os primeiros 18 bits para o Departamento de Engenharia Elétrica, obtemos 128.208.0.0 quando ANDing com a máscara de sub-rede. Isso corresponde ao endereço do prefixo, então o pacote é encaminhado para a interface que leva à rede de Engenharia Elétrica.

Página 470

446

A CAMADA DE REDE

INDIVÍDUO. 5

As divisões de sub-rede podem ser alteradas posteriormente, se necessário, atualizando todas as sub-redes

máscaras em roteadores dentro da universidade. Fora da rede, a sub-rede não é visível, portanto, alocar uma nova sub-rede não exige o contato com ICANN ou qualquer banco de dados externo.

CIDR - Classless InterDomain Routing

Mesmo que blocos de endereços IP sejam alocados para que os endereços sejam usados de forma confiável, ainda há um problema que permanece: explosão da tabela de roteamento. Roteadores em organizações na borda de uma rede, como uma universidade, precisam ter uma entrada para cada uma de suas sub-redes, informando ao roteador qual linha usar para chegar a essa rede. Para rotas para destinos fora da organização, eles

pode usar a regra padrão simples de enviar os pacotes na linha para o ISP que conecta a organização ao resto da Internet. O outro destino ad-

todos os vestidos devem estar por aí em algum lugar.

Roteadores em ISPs e backbones no meio da Internet não têm tais lux-
ury. Eles devem saber que caminho seguir para chegar a cada rede e não falha vai funcionar. Esses roteadores principais estão na **zona livre de padrão** do Internet. Ninguém sabe realmente quantas redes estão conectadas à Internet mais, mas é um grande número, provavelmente pelo menos um milhão. Isso pode fazer por uma mesa muito grande. Pode não parecer grande para os padrões do computador, mas perceba que os roteadores devem realizar uma pesquisa nesta tabela para encaminhar todos os pacotes, e os roteadores em grandes ISPs podem encaminhar até milhões de pacotes por segundo. Hard-
ware e memória rápida são necessários para processar pacotes nessas taxas, não um computador de propósito.

Além disso, os algoritmos de roteamento exigem que cada roteador troque informações sobre os endereços que ele pode alcançar com outros roteadores. Quanto maiores as tabelas, mais as informações precisam ser comunicadas e processadas. O processamento cresce em menos linearmente com o tamanho da mesa. Uma comunicação melhor aumenta a probabilidade que algumas partes se perderão, pelo menos temporariamente, possivelmente levando a instabilidades.

O problema da tabela de roteamento poderia ter sido resolvido indo para uma hierarquia mais profunda

archy, como a rede telefônica. Por exemplo, cada endereço IP contém um país, estado / província, cidade, rede e campo de host podem funcionar. Então, cada roteador só precisaria saber como chegar a cada país, os estados ou pro-
vinces em seu próprio país, as cidades em seu estado ou província e as redes em seu cidade. Infelizmente, esta solução exigiria consideravelmente mais do que 32 bits para endereços IP e usaria endereços de forma ineficiente (e Liechtenstein usaria têm tantos bits em seus endereços quanto os Estados Unidos).

Felizmente, há algo que podemos fazer para reduzir o tamanho da tabela de roteamento. Nós pode aplicar o mesmo insight da criação de sub-redes: roteadores em locais diferentes podem saber sobre um determinado endereço IP como pertencente a prefixos de tamanhos diferentes. No entanto, em vez de dividir um bloco de endereço em sub-redes, aqui combinamos vários pequenos

prefixos em um único prefixo maior. Este processo é denominado **agregação de rota**. o prefixo maior resultante é às vezes chamado de **super - rede**, para contrastar com as sub-redes como

a divisão de blocos de endereços.

Com a agregação, os endereços IP estão contidos em prefixos de tamanhos variados. o mesmo endereço IP que um roteador trata como parte de um / 22 (um bloco contendo 2¹⁰ ad-
vestidos) podem ser tratados por outro roteador como parte de um / 20 maior (que contém

2 12 endereços). Cabe a cada roteador ter o prefixo correspondente informado mação. Este projeto funciona com sub-redes e é chamado **CIDR** (**C**lassless **I**nter-**R**oteamento de Domínio), que é pronunciado como " cidra ", como na bebida. O mais recente sua versão é especificada no RFC 4632 (Fuller e Li, 2006). O nome destaca o contraste com endereços que codificam hierarquia com classes, que iremos descreba em breve.

Para tornar o CIDR mais fácil de entender, vamos considerar um exemplo em que um bloco de 8192 endereços IP está disponível a partir de 194.24.0.0. Suponha que Cambridge University precisa de 2.048 endereços e recebe os endereços 194.24.0.0 a 194.24.7.255, junto com a máscara 255.255.248.0. Este é um prefixo / 21. Próximo, A Universidade de Oxford pede 4096 endereços. Uma vez que um bloco de 4096 endereços deve residir em um limite de 4096 bytes, Oxford não pode receber endereços começando em 194.24.8.0. Em vez disso, obtém 194.24.16.0 a 194.24.31.255, junto com a sub-rede máscara 255.255.240.0. Finalmente, a Universidade de Edimburgo pede 1024 anúncios veste e recebe os endereços 194.24.8.0 a 194.24.11.255 e máscara 255.255.252.0. Essas atribuições estão resumidas na Figura 5.50.

Universidade
Primeiro endereço Último endereço

Quantos

Prefixo

Cambridge	
194.24.0.0	
194.24.7.255	
2048	
194.24.0.0/21	
Edimburgo	
194.24.8.0	
194.24.11.255	
1024	
194.24.8.0/22	
(Acessível)	
194.24.12.0	
194.24.15.255	
1024	
194.24.12.0/22	
Oxford	
194.24.16.0	
194.24.31.255	
4096	
194.24.16.0/20	

Figura 5-50. Um conjunto de atribuições de endereços IP.

Todos os roteadores na zona livre padrão agora são informados sobre os endereços IP nas três redes. Roteadores próximos às universidades podem precisar enviar uma linha de saída diferente para cada um dos prefixos, então eles precisam de uma entrada para cada um dos

prefixos em suas tabelas de roteamento. Um exemplo é o roteador em Londres na Figura 5.51.

Agora vamos olhar para essas três universidades do ponto de vista de um distante roteador em Nova York. Todos os endereços IP nos três prefixos devem ser enviados de Nova York (ou os EUA em geral) a Londres. O processo de roteamento em Londres percebe isso e combina os três prefixos em uma única entrada agregada para o prefixo 194.24.0.0/19 que passa para o roteador de Nova York. Este prefixo contém 8K aborda e cobre as três universidades e os 1024 anúncios de outra forma não alocados vestidos. Ao usar agregação, três prefixos foram reduzidos a um, reduzindo

192.24.16.0/20
192.24.0.0/21
Londres
Nova York
(3 prefixos)

Figura 5-51. Agregação de prefixos IP.

os prefixos sobre os quais o roteador de Nova York deve ser informado e a tabela de roteamento é tenta no roteador de Nova York.

Quando a agregação está ativada, é um processo automático. Isso depende de quais prefixos estão localizados na Internet e não nas ações de um administrador trator atribuindo endereços a redes. A agregação é muito usada em todo a Internet e pode reduzir o tamanho das tabelas do roteador para cerca de 200.000 prefixos. Como uma reviravolta adicional, os prefixos podem se sobrepor. A regra é que os pacotes são enviado na direção da rota mais específica ou do **prefixo correspondente mais longo** que tem o menor número de endereços IP. O roteamento de prefixo de correspondência mais longa fornece uma grau completo de flexibilidade, como visto no comportamento do roteador em Nova York em Fig. 5-52. Este roteador ainda usa um único prefixo agregado para enviar tráfego para o três universidades para Londres. No entanto, o bloco de anúncios previamente disponível vestidos com esse prefixo agora foram atribuídos a uma rede em San Francisco. Uma possibilidade é o roteador de Nova York manter quatro prefixos, enviando pacotes três deles para Londres e pacotes do quarto para San Francisco. Em vez de, o roteamento de prefixo de correspondência mais longa pode lidar com esse encaminhamento com os dois prefixos que são mostrados. Um prefixo geral é usado para direcionar o tráfego para todo o bloco para Londres. Mais um prefixo específico também é usado para direcionar uma parte do maior prefixo para San Francisco. Com a regra de prefixo correspondente mais longa, os endereços IP com na rede de São Francisco serão enviados na linha de saída para São Francisco, e todos os outros endereços IP no prefixo maior serão enviados para Londres. Conceitualmente, o CIDR funciona da seguinte maneira. Quando um pacote chega, o roteamento A tabela é verificada para determinar se o destino está dentro do prefixo. É possível que várias entradas com comprimentos de prefixo diferentes irão corresponder, caso em que a entrada com o prefixo mais longo é usada. Assim, se houver uma correspondência para uma máscara / 20 e uma máscara / 24, a entrada / 24 é usada para pesquisar a linha de saída do pacote. Como nunca, este processo seria tedioso se a tabela fosse realmente verificada entrada por entrada.

Página 473

SEC. 5,6
A CAMADA DE REDE NA INTERNET

449

192.24.0.0/19
192.24.8.0/22
192.24.16.0/20
192.24.0.0/21
Londres
Nova York
192.24.12.0/22
São Francisco
192.24.12.0/22

Figura 5-52. Roteamento de prefixo correspondente mais longo no roteador de Nova York.

Em vez disso, algoritmos complexos foram desenvolvidos para acelerar a correspondência de endereços

processo (Ruiz-Sanchez et al., 2001). Os roteadores comerciais usam chips VLSI personalizados com esses algoritmos embutidos no hardware.

Endereçamento Classful e Especial

Para ajudá-lo a compreender melhor por que o CIDR é tão útil, relacionaremos brevemente o design que o antecedeu. Antes de 1993, os endereços IP eram divididos em cinco categorias listadas na Figura 5-53. Esta alocação passou a ser chamada de **classful endereçamento**.

32 bits
Alcance do hospedeiro

```

endereços
1.0.0.0 a
127.255.255.255
128.0.0.0 a
191.255.255.255
192.0.0.0 a
223.255.255.255
224.0.0.0 a
239.255.255.255
240.0.0.0 a
255.255.255.255
Classe
0
Rede
Hospedeiro
10
Rede
Hospedeiro
110
Rede
Hospedeiro
1110
Endereço multicast
1111
Reservado para uso futuro
UMA
B
C
D
E

```

Figura 5-53. Formatos de endereço IP.

Os formatos de classe A, B e C permitem até 128 redes com 16 milhões hosts cada, 16.384 redes com até 65.536 hosts cada e 2 milhões de redes (por exemplo, LANs) com até 256 hosts cada (embora alguns deles sejam especiais). Além disso suportado é multicast (o formato de classe D), no qual um datagrama é direcionado para vários hosts. Os endereços que começam com 1111 são reservados para uso no futuro. Eles seriam valiosos para uso agora, devido ao esgotamento do espaço de endereços IPv4.

450

A CAMADA DE REDE INDIVÍDUO. 5

Infelizmente, muitos hosts não aceitam esses endereços como válidos porque eles estiveram fora dos limites por tanto tempo e é difícil ensinar novos truques aos antigos hosts. Este é um design hierárquico, mas ao contrário do CIDR, os tamanhos dos blocos de endereço são fixos. Existem mais de 2 bilhões de endereços, mas organizando o espaço de endereço por classes desperdiça milhões deles. Em particular, o verdadeiro vilão é a rede de classe B trabalhos. Para a maioria das organizações, uma rede de classe A, com 16 milhões de endereços, é muito grande, e uma rede de classe C, com 256 endereços é muito pequena. A classe B net-trabalho, com 65.536, está certo. No folclore da Internet, esta situação é conhecida como a **problema de três ursos** [como em *Cachinhos Dourados e os Três Ursos* (Southey, 1848)]. Na realidade, porém, um endereço de classe B é muito grande para a maioria das organizações. Estudos têm mostrado que mais da metade de todas as redes de classe B têm menos de 50 hospedeiros. Uma rede de classe C teria feito o trabalho, mas sem dúvida todas as organizações ção que pediu um endereço de classe B pensou que um dia superaria o 8-campo de host de bits. Em retrospecto, poderia ter sido melhor ter classe C net-funciona usa 10 bits em vez de 8 para o número do host, permitindo 1022 hosts por rede trabalhos. Se fosse esse o caso, a maioria das organizações provavelmente teria se conformado com uma rede de classe C, e haveria meio milhão delas (contra apenas 16.384 redes de classe B).

É difícil culpar os designers da Internet por não terem fornecido mais (e menores) endereços de classe B. Na época, foi tomada a decisão de criar os três aulas, a Internet era uma rede de pesquisa que conectava o principal centro de pesquisa cidades nos EUA (além de um número muito pequeno de empresas e instalações militares fazendo pesquisa em rede). Ninguém percebeu que a Internet estava se tornando uma massa sistema de comunicação de mercado que rivaliza com a rede telefônica. Na época, alguns

alguém sem dúvida disse: " Os Estados Unidos têm cerca de 2.000 faculdades e universidades. Mesmo se todos eles se conectam à Internet e muitas universidades em outros países aderem, também, nunca chegaremos a 16.000, uma vez que não há tantas universidades em o mundo inteiro. Além disso, tendo o número do host um número inteiro de bytes acelera o processamento de pacotes " (que então era feito inteiramente em software). Talvez algum dia as pessoas olhem para trás e culpem as pessoas que projetaram o televisor esquema de número de telefone e diga: " Que idiotas. Por que eles não incluíram o planeta o número do telefone? "“ Mas na época não parecia necessário. Para lidar com esses problemas, sub-redes foram introduzidas para atribuir blocos de maneira flexível de endereços dentro de uma organização. Mais tarde, o CIDR foi adicionado para reduzir o tamanho do a tabela de roteamento global. Hoje, os bits que indicam se um endereço IP deve ser longs para redes de classe A, B ou C não são mais usados, embora referências a estes aulas na literatura ainda são comuns.

Para ver como a eliminação das classes tornou o encaminhamento mais complicado, considere como era simples no antigo sistema de classes. Quando um pacote chega a um roteador, um a cópia do endereço IP foi deslocada 28 bits para a direita para gerar um número de classe de 4 bits. UMA

A ramificação de 16 vias classificou os pacotes em classes A, B, C (e D e E), com oito dos casos para a classe A, quatro dos casos para a classe B, e dois dos casos para classe C. O código para cada classe, em seguida, mascarado da rede de 8, 16 ou 24 bits

Página 475

SEC. 5,6

A CAMADA DE REDE NA INTERNET

451

número e alinhado à direita em uma palavra de 32 bits. O número da rede era então pesquisado na tabela A, B ou C, geralmente indexando para redes A e B e hashing para redes C. Assim que a entrada for encontrada, a linha de saída pode ser olhou para cima e o pacote foi encaminhado. Isso é muito mais simples do que o mais longo operação de prefixo correspondente, que não pode mais usar uma consulta de tabela simples porque um endereço IP pode ter qualquer prefixo de comprimento.

Os endereços de classe D continuam a ser usados na Internet para multicast. Na realidade, pode ser mais preciso dizer que eles estão começando a ser usados para multicast, já que o multicast da Internet não foi amplamente implantado no passado.

Existem também vários outros endereços que possuem significados especiais, conforme mostrado em

Fig. 5-54. O endereço IP 0.0.0.0, o endereço mais baixo, é usado pelos hosts quando eles estão sendo inicializados. Significa " esta rede " ou " este host. " Endereços IP com 0 como o número da rede se refere à rede atual. Esses endereços permitem máquinas para se referir à sua própria rede sem saber seu número (mas eles têm que saber a máscara de rede para saber quantos 0s incluir). O endereço consiste em todos 1s ou 255.255.255.255 - o endereço mais alto - é usado para significar todos os hosts na rede determinada. Ele permite a transmissão na rede local, normalmente uma LAN.

Os endereços com um número de rede adequado e todos os 1s no campo de host permitem machines para enviar pacotes de broadcast para LANs distantes em qualquer lugar na Internet. Como sempre, muitos administradores de rede desabilitam esse recurso, pois é principalmente uma segurança

perigo. Finalmente, todos os endereços do formulário 127. xx.yy.zz são reservados para loopback testando. Os pacotes enviados para esse endereço não são colocados na rede; eles são processados localmente e tratados como pacotes de entrada. Isso permite que os pacotes sejam enviados para

o host sem que o remetente saiba seu número, o que é útil para teste.

Este hospedeiro
Um host nesta rede
Transmitir no

```

rede local
0
Hospedeiro
Rede
127
(Qualquer coisa)
Transmitir em um
rede distante
Loopback
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
0 0
0 0
...
...
1111
1111

```

Figura 5-54. Endereços IP especiais.

NAT - tradução de endereço de rede

Os endereços IP são escassos. Um ISP pode ter um endereço / 16, dando a ele 65.534 números de host utilizáveis. Se tiver mais clientes do que isso, há um problema.

Página 476

452

A CAMADA DE REDE

INDIVÍDUO. 5

Essa escassez levou a técnicas para usar endereços IP com moderação. Um approach é atribuir dinamicamente um endereço IP a um computador quando ele está ligado e usando a rede, e para recuperar o endereço IP quando o host se torna ativa. O endereço IP pode então ser atribuído a outro computador que se torna ativa. Dessa forma, um único endereço / 16 pode lidar com até 65.534 usuários ativos. Esta estratégia funciona bem em alguns casos, por exemplo, para redes dial-up e computadores móveis e outros que podem estar temporariamente ausentes ou desligados. No entanto, ele não funciona muito bem para clientes empresariais. Muitos PCs em negócios esperam-se que os negócios estejam ativados continuamente. Alguns são máquinas de funcionários, apoiados acordados à noite, e alguns são servidores que podem ter que atender a uma solicitação remota em um aviso do momento. Essas empresas têm uma linha de acesso que sempre fornece conectividade com o resto da Internet.

Cada vez mais, esta situação também se aplica a usuários domésticos que assinam ADSL ou Internet por cabo, uma vez que não há cobrança de conexão (apenas uma taxa fixa mensal carregar). Muitos desses usuários têm dois ou mais computadores em casa, geralmente um para cada membro da família e todos querem estar online o tempo todo. A solução é para conectar todos os computadores em uma rede doméstica através de uma LAN e colocar um (sem fio) roteador nele. O roteador então se conecta ao ISP. Do ponto de vista do ISP, o a família agora é igual a uma pequena empresa com um punhado de computadores. Bem-vinda para Jones, Inc. Com as técnicas que vimos até agora, cada computador deve tem seu próprio endereço IP o dia todo. Para um ISP com muitos milhares de clientes, especialmente clientes empresariais e famílias que são como pequenas empresas Por outro lado, a demanda por endereços IP pode rapidamente exceder o bloco disponível. O problema de ficar sem endereços IP não é teórico que possa ocorrer em algum ponto no futuro distante. Isso está acontecendo aqui e agora.

A solução de longo prazo é que toda a Internet migre para IPv6, que tem Endereços de 128 bits. Essa transição está ocorrendo lentamente, mas vai se passar anos antes o processo está completo. Para sobreviver enquanto isso, uma solução rápida era necessária. A solução rápida que é amplamente usada hoje veio na forma de **NAT (Endereço de Rede Tradução)**, que é descrito no RFC 3022 e que iremos resumir abaixo. Para obter informações adicionais, consulte Dutcher (2001). A ideia básica por trás do NAT é que o ISP atribua a cada residência ou empresa um endereço IP único (ou, no máximo, um pequeno número deles) para tráfego da Internet. Dentro

a rede do cliente, cada computador recebe um endereço IP exclusivo, que é usado para roteamento de tráfego intramural. No entanto, pouco antes de um pacote sair da rede do cliente funciona e vai para o ISP, uma tradução de endereço do anúncio IP interno exclusivo vestido para o endereço IP público compartilhado ocorre. Esta tradução faz uso de três intervalos de endereços IP que foram declarados como privados. Redes podem usar-se internamente como quiserem. A única regra é que nenhum pacote contendo esses endereços podem aparecer na própria Internet. Os três intervalos reservados são:

10.0.0
- 10.255.255.255/8
(16.777.216 hosts)
172.16.0.0 - 172.31.255.255/12
(1.048.576 hosts)
192.168.0.0 - 192.168.255.255/16 (65.536 hosts)

Página 477

SEC. 5,6
A CAMADA DE REDE NA INTERNET

453

O primeiro intervalo fornece 16.777.216 endereços (exceto para todos os 0s e todos os 1s, como usual) e é a escolha usual, mesmo se a rede não for grande.

A operação do NAT é mostrada na Figura 5-55. Dentro das instalações do cliente, cada máquina possui um endereço exclusivo no formato 10. xyz . No entanto, antes de um pacote sair das instalações do cliente, ele passa por uma **caixa NAT** que converte o endereço IP de origem ternal, 10.0.0.1 na figura, para o verdadeiro endereço IP do cliente, 198.60.42.12 neste exemplo. A caixa NAT é frequentemente combinada em um único dispositivo com um firewall, que fornece segurança controlando cuidadosamente o que entra a rede do cliente e o que sai dela. Estudaremos firewalls no Cap.

8. Também é possível integrar a caixa NAT em um roteador ou modem ADSL.

Pacote após
tradução
Limite das instalações do cliente
Caixa NAT / firewall
ISP
roteador
IP = 198.60.42.12
porta = 3344
IP = 10.0.0.1
porta = 5544
(para Internet)
Pacote antes
tradução
Cliente
roteador
e LAN

Figura 5-55. Colocação e funcionamento de uma caixa NAT.

Até agora, omitimos um pequeno, mas crucial detalhe: quando a resposta vem de volta (por exemplo, de um servidor Web), é naturalmente endereçado a 198.60.42.12, então como a caixa NAT sabe por qual endereço interno deve ser substituída? Aqui está o problema com o NAT. Se houvesse um campo sobressalente no cabeçalho IP, esse campo poderia ser

usado para rastrear quem era o remetente real, mas apenas 1 bit ainda não foi usado. No princípio, uma nova opção poderia ser criada para manter o verdadeiro endereço de origem, mas fazendo

por isso, seria necessário alterar o código IP em todas as máquinas em toda a Internet para lidar com a nova opção. Esta não é uma alternativa promissora para uma solução rápida. O que realmente acontece é o seguinte. Os designers do NAT observaram que a maioria dos pacotes IP transportam cargas úteis TCP ou UDP. Quando estudamos TCP e UDP em Indivíduo. 6, veremos que ambos têm cabeçalhos contendo uma porta de origem e um Porto de destino. Abaixo, discutiremos apenas as portas TCP, mas exatamente a mesma história vale para portas UDP. As portas são inteiros de 16 bits que indicam onde o TCP a conexão começa e termina. Essas portas fornecem o campo necessário para fazer o NAT trabalhos.

Quando um processo deseja estabelecer uma conexão TCP com um processo remoto, ele

conecta-se a uma porta TCP não utilizada em sua própria máquina. Isso é chamado de **porta de origem** e diz ao código TCP para onde enviar os pacotes de entrada pertencentes ao esta conexão. O processo também fornece uma **porta de destino** para dizer a quem dar

454

A CAMADA DE REDE

INDIVÍDUO. 5

os pacotes no lado remoto. As portas de 0 a 1023 são reservadas para servidores bem conhecidos vícios. Por exemplo, a porta 80 é a porta usada por servidores Web, para que clientes remotos possam

localize-os. Cada mensagem TCP de saída contém uma porta de origem e um destino porto da nação. Juntas, essas portas servem para identificar os processos usando o conexão em ambas as extremidades.

Uma analogia pode tornar o uso de portas mais claro. Imagine uma empresa com um único número de telefone principal. Quando as pessoas ligam para o número principal, elas alcançam um

operadora que pergunta qual ramal eles querem e os encaminha para aquele extensão. O número principal é análogo ao endereço IP do cliente e ao extensões em ambas as extremidades são análogas às portas. As portas são efetivamente um extra 16 bits de endereçamento que identificam qual processo obtém qual pacote de entrada.

Usando o campo *Porta de origem*, podemos resolver nosso problema de mapeamento. Sempre que um pacote de saída entra na caixa NAT, o endereço de origem 10. xyz é substituído por o verdadeiro endereço IP do cliente. Além disso, o campo *Porta de origem* TCP é substituído por um índice na tabela de conversão de 65.536 entradas da caixa NAT. Esta entrada da tabela contém o endereço IP original e a porta de origem original. Finalmente, tanto o IP e as somas de verificação do cabeçalho TCP são recalculadas e inseridas no pacote. Isto é necessário substituir a *porta de origem* porque as conexões das máquinas 10.0.0.1 e 10.0.0.2 podem usar a porta 5000, por exemplo, então a *porta de origem* por si só não é suficiente para identificar o processo de envio.

Quando um pacote chega na caixa NAT do ISP, a *porta de origem* no O cabeçalho TCP é extraído e usado como um índice na tabela de mapeamento da caixa NAT.

A partir da entrada localizada, o endereço IP interno e a porta de *origem* TCP original são extraído e inserido no pacote. Então, as somas de verificação IP e TCP são recomputado e inserido no pacote. O pacote é então passado para o cliente roteador mer para entrega normal usando o endereço 10. xyz .

Embora esse esquema resolva o problema, os puristas da rede no IP comunidade tem uma tendência a considerá-lo uma abominação-na-face-da-terra. Resumidamente, aqui estão algumas das objeções. Primeiro, o NAT viola o modelo arquitetônico de IP, que afirma que cada endereço IP identifica de forma única localiza uma única máquina em todo o mundo. Toda a estrutura de software da Internet é construído sobre este fato. Com o NAT, milhares de máquinas podem (e usam) o endereço 10.0.0.1.

Em segundo lugar, o NAT quebra o modelo de conectividade ponta a ponta da Internet, que diz que qualquer host pode enviar um pacote a qualquer outro host a qualquer momento. Desde o mapa

ping na caixa de NAT é configurado por pacotes de saída, pacotes de entrada não podem ser aceito até depois dos de saída. Na prática, isso significa que um usuário doméstico com O NAT pode fazer conexões TCP / IP com um servidor Web remoto, mas um usuário remoto não pode fazer conexões com um servidor de jogos na rede doméstica. Configuração especial ração ou técnicas de **passagem de NAT** são necessárias para suportar esse tipo de situação.

Terceiro, o NAT muda a Internet de uma rede sem conexão para um peculiar tipo de rede orientada para conexão. O problema é que a caixa NAT deve manter informações (ou seja, o mapeamento) para cada conexão que passa por ele.

455

Fazer com que a rede mantenha o estado de conexão é uma propriedade orientada à conexão redes, não aquelas sem conexão. Se a caixa NAT travar e sua tabela de mapeamento é perdido, todas as suas conexões TCP são destruídas. Na ausência de NAT, um roteador pode travar e reiniciar sem efeito de longo prazo nas conexões TCP. O processo de envio Ess apenas expira em alguns segundos e retransmite todos os pacotes não confirmados ets. Com o NAT, a Internet se torna tão vulnerável quanto uma rede comutada por circuito.

Quarto, o NAT viola a regra mais fundamental de camadas de protocolo: camada k pode não fazer suposições sobre qual camada $k + 1$ colocou na carga útil campo. Este princípio básico existe para manter as camadas independentes. Se TCP for posterior atualizado para TCP-2, com um layout de cabeçalho diferente (por exemplo, portas de 32 bits), o NAT irá

falhou. A ideia dos protocolos em camadas é garantir que as mudanças em uma camada não requer mudanças em outras camadas. O NAT destrói essa independência.

Quinto, os processos na Internet não precisam usar TCP ou UDP. Se um usuário na máquina A decide usar algum novo protocolo de transporte para falar com um usuário na máquina B (por exemplo, para uma aplicação multimídia), introdução de uma caixa NAT fará com que o aplicativo falhe porque a caixa NAT não será capaz de localizar a porta de *origem* TCP corretamente.

Um sexto problema relacionado é que alguns aplicativos usam múltiplos TCP / IP conexões ou portas UDP de maneiras prescritas. Por exemplo, **FTP**, o padrão

Protocolo de transferência de arquivos, insere endereços IP no corpo do pacote para o receptor para extrair e usar. Como o NAT não sabe nada sobre esses arranjos, ele não pode reescrever os endereços IP ou de outra forma contabilizá-los. Esta falta de em pé significa que o FTP e outros aplicativos, como o H.323 Internet tele-protocolo falso (que estudaremos no Capítulo 7) falhará na presença de NAT a menos que precauções especiais sejam tomadas. Muitas vezes, é possível corrigir o NAT para esses

casos, mas ter que corrigir o código na caixa NAT toda vez que um novo aplicativo aparecer não é uma boa ideia.

Finalmente, como o campo da *porta de origem* TCP é de 16 bits, no máximo 65.536 máquinas pode ser mapeado em um endereço IP. Na verdade, o número é um pouco menor porque as primeiras 4096 portas são reservadas para usos especiais. No entanto, se vários anúncios de IP vestidos estão disponíveis, cada um com capacidade para 61.440 máquinas.

Uma visão desses e de outros problemas com o NAT é fornecida na RFC 2993. Apesar problemas, o NAT é amplamente utilizado na prática, especialmente para casa e pequenas empresas redes, como a única técnica expediente para lidar com a escassez de endereços IP. isto envolveu-se com firewalls e privacidade porque bloqueia não solicitados pacotes de entrada por padrão. Por este motivo, é improvável que desapareça, mesmo quando IPv6 é amplamente implantado.

5.6.3 IP Versão 6

IP tem sido muito usado por décadas. Funcionou extremamente bem, pois demonstrado pelo crescimento exponencial da Internet. Infelizmente, o IP foi- tornou-se vítima de sua própria popularidade: seus endereços estão quase acabando. Até

456

com CIDR e NAT usando endereços com mais moderação, os últimos endereços IPv4 são deverá ser atribuído pela ICANN antes do final de 2012. Este desastre iminente foi reconhecido há quase duas décadas, e gerou muita discussão e polêmica na comunidade da Internet sobre o que fazer a respeito.

Nesta seção, iremos descrever o problema e várias soluções propostas ções. A única solução de longo prazo é mudar para endereços maiores. **IPv6** (IP ver-

seção 6) é um projeto de substituição que faz exatamente isso. Ele usa endereços de 128 bits; uma a falta desses endereços não é provável em um futuro previsível. Como-nunca, o IPv6 provou ser muito difícil de implantar. É uma camada de rede diferente protocolo que realmente não funciona em conjunto com o IPv4, apesar de muitas semelhanças. Além disso,

as empresas e os usuários não sabem ao certo por que deveriam querer o IPv6 em qualquer caso. O resultado é que o IPv6 é implantado e usado em apenas uma pequena fração da Internet (as estimativas são de 1%), apesar de ter sido um padrão da Internet desde 1998. O próximo vários anos será um período interessante, pois os poucos endereços IPv4 restantes são alocado. As pessoas começarão a leiloar seus endereços IPv4 no eBay? Será um mercado negro neles surge? Quem sabe.

Além dos problemas de endereço, outras questões surgem em segundo plano. No nos primeiros anos, a Internet foi amplamente usada por universidades, indústrias de alta tecnologia, e o Governo dos Estados Unidos (especialmente o Departamento de Defesa). Com a explosão de interesse na Internet a partir de meados da década de 1990, começou a ser usado por uma grupo diferente de pessoas, geralmente com requisitos diferentes. Por um lado, numerosas pessoas com smartphones usam-no para se manterem em contato com suas bases. Para outro, com a convergência iminente do computador, comunicação e indústrias de entretenimento, pode não demorar muito para que todos os telefones visão definida no mundo é um nó da Internet, resultando em um bilhão de máquinas sendo usado para áudio e vídeo sob demanda. Nessas circunstâncias, tornou-se aparente que o IP teve que evoluir e se tornar mais flexível.

Vendo esses problemas no horizonte, em 1990 a IETF começou a trabalhar em um novo versão do IP, que nunca ficaria sem endereços, resolveria uma variedade de outros problemas, e ser mais flexível e eficiente também. Seus principais objetivos eram:

1. Suporta bilhões de hosts, mesmo com alocação de endereços ineficiente.
2. Reduza o tamanho das tabelas de roteamento.
3. Simplifique o protocolo para permitir que os roteadores processem os pacotes com mais rapidez.
4. Proporcionar melhor segurança (autenticação e privacidade).
5. Preste mais atenção ao tipo de serviço, especialmente para dados em tempo real.
6. Ajude a multidifusão permitindo que os escopos sejam especificados.
7. Torne possível que um host faça roaming sem alterar seu endereço.
8. Permita que o protocolo evolua no futuro.
9. Permitir que o protocolo antigo e o novo coexistam por anos.

Página 481

SEC. 5,6

A CAMADA DE REDE NA INTERNET

457

O design do IPv6 apresentou uma grande oportunidade para melhorar todos os recursos características no IPv4 que ficam aquém do que é desejado agora. Para desenvolver um protocolo que atendesse todos esses requisitos, a IETF emitiu uma chamada para propostas e discussão na RFC 1550. Vinte e uma respostas foram inicialmente recebidas. Em dezembro de 1992, sete propostas sérias estavam sobre a mesa. Eles variaram de fazer pequenos remendos a IP, para jogá-lo fora por completo e substituí-lo por um proto completamente diferente col.

Uma proposta era executar TCP sobre CLNP, o protocolo da camada de rede de-assinado para OSI. Com seus endereços de 160 bits, o CLNP teria fornecido endereçar o espaço para sempre, pois poderia dar a cada molécula de água nos oceanos endereços suficientes (aproximadamente 2⁵⁰) para configurar uma pequena rede. Esta escolha também

unificaram dois protocolos de camada de rede principais. No entanto, muitas pessoas sentiram que isso teria sido uma admissão de que algo no mundo OSI era realmente bem feito, uma afirmação considerada politicamente incorreta nos círculos da Internet. CLNP foi padronizado de perto no IP, então os dois não são realmente tão diferentes. Na verdade, o protocolo escolhido difere do IP muito mais do que o CLNP. Outro

A greve contra o CLNP foi seu fraco suporte para tipos de serviço, algo necessário para transmitir multimídia de forma eficiente.

Três das melhores propostas foram publicadas na *Rede IEEE* (Deering, 1993; Francis, 1993; e Katz e Ford, 1993). Depois de muita discussão, revisão, e disputando a posição, uma versão combinada modificada do Deering e As propostas de Francis, agora chamadas de **SIPP** (**Simple Internet Protocol Plus**) foram selecionadas

selecionou e recebeu a designação **IPv6**.

O IPv6 atende às metas da IETF muito bem. Ele mantém as boas características do IP, descartões ou não enfatiza os ruins e adiciona novos quando necessário. Em geral, IPv6 não é compatível com IPv4, mas é compatível com o outro auxiliar Protocolos de Internet, incluindo TCP, UDP, ICMP, IGMP, OSPF, BGP e DNS, com pequenas modificações sendo necessárias para lidar com endereços mais longos. O principal os recursos do IPv6 são discutidos a seguir. Mais informações sobre isso podem ser encontradas em RFCs 2460 a 2466.

Em primeiro lugar, o IPv6 tem endereços mais longos do que o IPv4. Eles são 128 bits longo, que resolve o problema que o IPv6 se propôs a resolver: fornecer um eficaz fornecimento ilimitado de endereços de Internet. Teremos mais a dizer sobre os endereços Em breve.

A segunda grande melhoria do IPv6 é a simplificação do cabeçalho. isto contém apenas sete campos (contra 13 no IPv4). Esta mudança permite que os roteadores processar pacotes mais rápido e, assim, melhorar o rendimento e o atraso. Vamos discutir o cabeçalho em breve também.

A terceira grande melhoria é um melhor suporte para opções. Esta mudança foi essencial com o novo cabeçalho porque os campos que antes eram obrigatórios são agora opcional (porque eles não são usados com tanta frequência). Além disso, as opções de caminho são representados é diferente, tornando simples para os roteadores pularem opções não destinado a eles. Este recurso acelera o tempo de processamento do pacote.

Página 482

458

A CAMADA DE REDE INDIVÍDUO. 5

Uma quarta área em que o IPv6 representa um grande avanço é a segurança. IETF teve sua quantidade de notícias de jornal sobre crianças de 12 anos precoces usando seus computadores para invadir bancos e bases militares em toda a Internet. Houve um forte sentimento de que algo precisava ser feito para melhorar a segurança. Autenticação e privacidade são os principais recursos do novo IP. Estes foram posteriormente adaptados para IPv4, entretanto, na área de segurança as diferenças não são mais tão grandes.

Finalmente, mais atenção foi dada à qualidade do serviço. Vários meios-esforços sinceros para melhorar a QoS foram feitos no passado, mas agora, com o crescimento da multimídia na Internet, o senso de urgência é maior.

O cabeçalho IPv6 principal

O cabeçalho IPv6 é mostrado na Figura 5.56. O campo *Versão* é sempre 6 para IPv6 (e 4 para IPv4). Durante o período de transição do IPv4, que já demorou mais de uma década, os roteadores serão capazes de examinar este campo para dizer que tipo de pacote que eles têm. Como um aparte, fazer este teste desperdiça algumas instruções no caminho crítico, visto que o cabeçalho do link de dados geralmente indica o protocolo de rede para demultiplexação, portanto, alguns roteadores podem ignorar a verificação. Por exemplo, o Ether-

O campo *tipo de rede* tem valores diferentes para indicar uma carga útil IPv4 ou IPv6. o discussões entre os campos " Faça certo " e " Faça rápido " serão, sem dúvida, longo e vigoroso.

32 bits
Versão
Etiqueta de fluxo
Diff. Serviços
Próximo cabeçalho

Comprimento da carga útil
Limite de salto
Endereço de Origem
(16 bytes)
Endereço de destino
(16 bytes)

Figura 5-56. O cabeçalho fixo IPv6 (obrigatório).

O campo de *serviços diferenciados* (originalmente chamado de *classe de tráfego*) é usado para distinguir a classe de serviço para pacotes com entrega em tempo real diferente

Página 483

SEC. 5,6
A CAMADA DE REDE NA INTERNET

459

requisitos. É utilizado com arquitetura de serviço diferenciada pela qualidade de serviço da mesma maneira que o campo de mesmo nome no pacote IPv4. Além disso, os 2 bits de baixa ordem são usados para sinalizar indicações explícitas de congestionamento, novamente no

da mesma forma que com IPv4.

O campo de *rótulo de fluxo* fornece uma maneira para uma fonte e um destino marcarem grupos de pacotes que têm os mesmos requisitos e devem ser tratados no

da mesma forma pela rede, formando uma pseudoconexão. Por exemplo, um fluxo de pacotes de um processo em um determinado host de origem para um processo em um destino específico

host da nação pode ter requisitos de atraso rigorosos e, portanto, precisa de banda reservada largura. O fluxo pode ser configurado com antecedência e receber um identificador. Quando um pacote

com um *rótulo de fluxo* diferente de zero exibido, todos os roteadores podem procurá-lo em tabelas para ver que tipo de tratamento especial requer. Na verdade, os fluxos são um atentamos ter as duas coisas: a flexibilidade de uma rede de datagramas e a garantia

tees de uma rede de circuito virtual.

Cada fluxo para fins de qualidade de serviço é designado pelo endereço de origem, endereço de destino e número do fluxo. Este projeto significa que até 2²⁰ fluxos pode estar ativo ao mesmo tempo entre um determinado par de endereços IP. Isso também significa que mesmo se dois fluxos vindos de hosts diferentes, mas com o mesmo fluxo etiqueta passar pelo mesmo roteador, o roteador será capaz de distingui-los usando os endereços de origem e destino. Espera-se que os rótulos de fluxo sejam escolhidos sen aleatoriamente, em vez de atribuído sequencialmente começando em 1, então os roteadores são ex-

pected para hash eles.

O campo de *comprimento da carga útil* informa quantos bytes seguem o cabeçalho de 40 bytes de Fig. 5-56. O nome do campo *Comprimento total* IPv4 foi alterado porque o significado foi ligeiramente alterado: os 40 bytes do cabeçalho não são mais contados como parte do comprimento (como costumavam ser). Essa mudança significa que a carga útil agora pode ser 65.535 bytes em vez de meros 65.515 bytes.

O próximo campo de *cabeçalho* deixa o gato fora da bolsa. A razão pela qual o cabeçalho poderia ser simplificado é que pode haver cabeçalhos de extensão adicionais (opcionais). Isto campo informa qual dos (atualmente) seis cabeçalhos de extensão, se houver, segue este.

Se este cabeçalho for o último cabeçalho IP, o próximo campo de *cabeçalho* informa qual transporte pro

manipulador de tocol (por exemplo, TCP, UDP) para o qual passar o pacote.

O campo *Hop limit* é usado para impedir que os pacotes durem para sempre. É, em prática, o mesmo que o campo *Time to live* no IPv4, ou seja, um campo que é decrementado em cada salto. Em teoria, no IPv4 era um tempo em segundos, mas nenhum roteador usou forma, então o nome foi alterado para refletir a forma como é realmente usado.

Em seguida, vêm os campos *Endereço de origem* e *Endereço de destino*. Deering's a proposta original, SIP, usava endereços de 8 bytes, mas durante o processo de revisão muitos as pessoas achavam que, com endereços de 8 bytes, o IPv6 ficaria sem endereços em um poucas décadas, ao passo que com endereços de 16 bytes ele nunca acabaria. Outras pessoas

ple argumentou que 16 bytes era um exagero, enquanto outros ainda favoreciam o uso de 20 bytes endereços para serem compatíveis com o protocolo de datagrama OSI. Ainda outra facção endereços de tamanho variável desejados. Depois de muito debate e mais do que algumas palavras

Página 484

460

A CAMADA DE REDE

INDIVÍDUO. 5

não imprimível em um livro acadêmico, foi decidido que o anúncio de 16 bytes de comprimento fixo vestidos eram o melhor compromisso.

Uma nova notação foi desenvolvida para escrever endereços de 16 bytes. Eles são escrito como oito grupos de quatro dígitos hexadecimais com dois pontos entre os grupos, como isso:

8000: 0000: 0000: 0000: 0123: 4567: 89AB: CDEF

Uma vez que muitos endereços terão muitos zeros dentro deles, três otimizações têm sido autorizado. Primeiro, zeros à esquerda dentro de um grupo podem ser omitidos, então 0123 pode ser escrito como 123. Em segundo lugar, um ou mais grupos de 16 bits zero podem ser substituídos por

um par de dois pontos. Assim, o endereço acima agora se torna

8000 :: 123: 4567: 89AB: CDEF

Finalmente, os endereços IPv4 podem ser escritos como um par de dois pontos e um antigo ponto número decimal, por exemplo:

:: 192.31.20.46

Talvez seja desnecessário ser tão explícito sobre isso, mas existem muitos 16-endereços de bytes. Especificamente, existem 2^{128} deles, o que é aproximadamente 3×10^{38} . Se toda a terra, terra e água fossem cobertas por computadores, IPv6 permitiria 7×10^{23} endereços IP por metro quadrado. Estudantes de química vão observar que esse número é maior do que o número de Avogadro. Embora não fosse a intenção de dar a cada molécula na superfície da Terra seu próprio endereço IP, nós não estamos tão longe.

Na prática, o espaço de endereço não será usado de forma eficiente, assim como o telefone o espaço de endereço do número não (o código de área de Manhattan, 212, está quase cheio, mas que para Wyoming, 307, está quase vazio). Na RFC 3194, Durand e Huitema calculinou nisso, usando a atribuição de números de telefone como guia, mesmo no cenário mais pessimista, ainda haverá bem mais de 1000 endereços IP por metros quadrados de toda a superfície terrestre (terra e água). Em qualquer cenário provável, haverá trilhões deles por metro quadrado. Em suma, parece improvável que nós acabará em um futuro previsível.

É instrutivo comparar o cabeçalho IPv4 (Fig. 5-46) com o cabeçalho IPv6 (Figura 5-56) para ver o que foi omitido no IPv6. O campo *IHL* se foi porque o cabeçalho IPv6 tem um comprimento fixo. O campo *Protocolo* foi removido porque o Próximo campo de *cabeçalho* informa o que segue o último cabeçalho de IP (por exemplo, um segmento UDP ou TCP).

Todos os campos relacionados à fragmentação foram removidos porque o IPv6 leva uma abordagem diferente à fragmentação. Para começar, todos os hosts em conformidade com IPv6 são deve determinar dinamicamente o tamanho do pacote a ser usado. Eles fazem isso usando o procedimento de descoberta de MTU de caminho que descrevemos na Seç. 5.5.5. Em resumo, quando um host

envia um pacote IPv6 muito grande, em vez de fragmentá-lo, o roteador que é incapaz de encaminhar, ele descarta o pacote e envia uma mensagem de erro de volta para o

Página 485

SEC. 5,6

A CAMADA DE REDE NA INTERNET

461

envio de host. Esta mensagem diz ao host para quebrar todos os pacotes futuros para aquele destino. Fazer com que o host envie pacotes com o tamanho certo em primeiro lugar é, em última análise, muito mais eficiente do que ter os roteadores fragmentá-los no meio. Além disso, o pacote de tamanho mínimo que os roteadores devem ser capazes de encaminhar foi aumentado de 576 para 1280 bytes para permitir 1024 bytes de dados e muitos cabeçalhos. Finalmente, o campo *Checksum* foi embora porque calculá-lo reduz muito a performance. Com as redes confiáveis agora utilizadas, combinado com o fato de que a camada de enlace de dados e camadas de transporte normalmente têm suas próprias somas de verificação, o valor de mais uma soma de verificação não valeu o preço de desempenho extraído. A remoção de todos esses recursos resultou em uma rede enxuta e média protocolo de camada. Assim, o objetivo do IPv6 - um protocolo rápido, mas flexível, com muito de espaço de endereço - é atendido por este design.

Cabeçalhos de extensão

Alguns dos campos IPv4 ausentes ocasionalmente ainda são necessários, então o IPv6 introduz o conceito de **cabeçalhos de extensão** (opcionais). Esses cabeçalhos podem ser substituídos ou fornecer informações extras, mas codificadas de forma eficiente. Seis tipos de cabeçalhos de extensão são definidos no momento, conforme listado na Figura 5.57. Cada um é opcional, mas se mais de um estiver presente, eles devem aparecer diretamente após o cabeçalho e, de preferência, na ordem listada.

Cabeçalho de extensão

Descrição

- Opções de salto a salto
- Informações diversas para roteadores
- Opções de destino
- Informações adicionais para o destino
- Encaminhamento
- Lista perdida de roteadores para visitar
- Fragmentação
- Gerenciamento de fragmentos de datagrama
- Autenticação
- Verificação da identidade do remetente
- Carga útil de segurança criptografada
- Informações sobre o conteúdo criptografado

Figura 5-57. Cabeçalhos de extensão IPv6.

Alguns dos cabeçalhos têm um formato fixo; outros contêm um número variável de opções de comprimento variável. Para estes, cada item é codificado como um (*tipo, comprimento, Valor*) tupla. O *Tipo* é um campo de 1 byte que informa qual opção é essa. Os valores foram escolhidos de modo que os primeiros 2 bits digam aos roteadores que não sabem como processar a opção o que fazer. As opções são: ignorar a opção; descartar o pacote; descarte o pacote e envie de volta um pacote ICMP; e descarte o pacote mas não envie pacotes ICMP para endereços multicast (para evitar um multílico de geração de milhões de relatórios ICMP).

O *comprimento* também é um campo de 1 byte. Diz quanto tempo o valor é (0 a 255 bytes). O *valor* é qualquer informação necessária, até 255 bytes.

Página 486

462

A CAMADA DE REDE

INDIVÍDUO. 5

O cabeçalho hop-by-hop é usado para informações de que todos os roteadores ao longo do caminho deve examinar. Até agora, uma opção foi definida: suporte de datagramas excedendo 64 KB. O formato desse cabeçalho é mostrado na Figura 5.58. Quando é usado, o campo de *comprimento de carga útil* no cabeçalho fixo é definido como 0.

Próximo cabeçalho
Comprimento da carga útil Jumbo
0
194
4

Figura 5-58. O cabeçalho de extensão salto a salto para datagramas grandes (jumbogramas).

Tal como acontece com todos os cabeçalhos de extensão, este começa com um byte informando que tipo de

cabeçalho vem a seguir. Este byte é seguido por outro que informa quanto tempo o salto a salto o cabeçalho está em bytes, excluindo os primeiros 8 bytes, que são obrigatórios. Tudo extenso sões começam dessa maneira.

Os próximos 2 bytes indicam que esta opção define o tamanho do datagrama (código 194) e que o tamanho é um número de 4 bytes. Os últimos 4 bytes fornecem o tamanho dos dados grama. Tamanhos menores que 65.536 bytes não são permitidos e resultarão no primeiro roteador descartando o pacote e enviando de volta uma mensagem de erro ICMP. Dados-gramas que usam essa extensão de cabeçalho são chamados de **jumbogramas**. O uso de jumbo-gramas é importante para aplicativos de supercomputador que devem transferir gigabytes de dados de forma eficiente na Internet.

O cabeçalho de opções de destino é destinado a campos que precisam apenas ser interpretada no host de destino. Na versão inicial do IPv6, as únicas opções defined são opções nulas para preencher este cabeçalho em um múltiplo de 8 bytes, então inicialmente não será usado. Foi incluído para garantir que o novo roteamento e host

o software pode lidar com isso, caso alguém pense em uma opção de destino algum dia.

O cabeçalho de roteamento lista um ou mais roteadores que devem ser visitados no caminho para o destino. É muito semelhante ao roteamento de origem livre IPv4 em que todos os anúncios vestidos listados devem ser visitados em ordem, mas outros roteadores não listados podem ser visitados

entre. O formato do cabeçalho de roteamento é mostrado na Figura 5.59.

Próximo cabeçalho

Extensão de cabeçalho

comprimento

Tipo de roteamento

Segmentos restantes

Dados específicos do tipo

Figura 5-59. O cabeçalho da extensão para roteamento.

Página 487

SEC. 5,6

A CAMADA DE REDE NA INTERNET

463

Os primeiros 4 bytes do cabeçalho da extensão de roteamento contêm quatro inteiros de 1 byte.

Os campos *Next header* e *Header extension length* foram descritos acima. o

O campo do *tipo de roteamento* fornece o formato do restante do cabeçalho. Tipo 0 diz que um re-a palavra de 32 bits servida segue a primeira palavra, seguida por algum número de anúncios IPv6 vestidos. Outros tipos podem ser inventados no futuro, conforme necessário. Finalmente, o *Seg-o campo esquerdo de comentários* mantém o controle de quantos dos endereços da lista ainda não foram visitados. É diminuído toda vez que um é visitado. Quando chega a 0, o

o pacote está sozinho, sem mais orientações sobre a rota a seguir. Geralmente,

neste ponto, está tão perto do destino que o melhor caminho é óbvio.

O cabeçalho do fragmento lida com a fragmentação de maneira semelhante ao IPv4.

O cabeçalho contém o identificador do datagrama, o número do fragmento e um bit revelador se mais fragmentos virão. No IPv6, ao contrário do IPv4, apenas o host de origem pode fragmentar um pacote. Os roteadores ao longo do caminho podem não fazer isso. Esta mudança é uma

grande ruptura filosófica com o IP original, mas de acordo com a prática atual tice para IPv4. Além disso, simplifica o trabalho dos roteadores e torna o roteamento mais rápido. Como

mencionado acima, se um roteador for confrontado com um pacote que é muito grande, ele descarta o pacote e envia um pacote de erro ICMP de volta à origem. Essa informação permite ao host de origem fragmentar o pacote em pedaços menores usando este cabeçalho e tente novamente.

O cabeçalho de autenticação fornece um mecanismo pelo qual o receptor de um pacote pode ter certeza de quem o enviou. A carga útil de segurança criptografada torna possível capaz de criptografar o conteúdo de um pacote para que apenas o destinatário pretendido possa Leia-o. Esses cabeçalhos usam as técnicas criptográficas que iremos descrever em

Indivíduo. 8 para cumprir suas missões.

Controvérsias

Dado o processo de design aberto e as opiniões fortemente defendidas de muitos dos pessoas envolvidas, não deve ser surpresa que muitas escolhas feitas para IPv6 foram altamente controversos, para dizer o mínimo. Vamos resumir alguns desses brevemente abaixo. Para todos os detalhes sangrentos, consulte os RFCs.

Já mencionamos o argumento sobre o comprimento do endereço. O resultado era um meio-termo: endereços de comprimento fixo de 16 bytes.

Outra luta se desenvolveu ao longo do campo de *limite de Hop*. Um acampamento sentiu fortemente que limitar o número máximo de saltos a 255 (implícito no uso de um Campo de 8 bits) foi um erro grosseiro. Afinal, caminhos de 32 saltos são comuns agora, e Daqui a 10 anos, caminhos muito mais longos podem ser comuns. Essas pessoas argumentaram que usar um tamanho de endereço enorme era previdente, mas usar uma contagem de saltos minúscula era curto-

avistado. Em sua opinião, o maior pecado que um cientista da computação pode cometer é vide poucos bits em algum lugar.

A resposta foi que os argumentos poderiam ser feitos para aumentar cada campo, levando ing para um cabeçalho inchado. Além disso, a função do campo de *limite de salto* é manter o pacote ets de vagar por muito tempo e 65.535 saltos é muito, muito tempo.

Página 488

464

A CAMADA DE REDE

INDIVÍDUO. 5

Finalmente, conforme a Internet cresce, mais e mais links de longa distância serão construídos, tornando possível ir de qualquer país para qualquer outro país em meia dúzia lúpulo, no máximo. Se demorar mais de 125 saltos para ir da fonte e do destino nação aos seus respectivos portais internacionais, algo está errado com o na-backbones tradicionais. Os 8 bits ganharam este.

Outra batata quente era o tamanho máximo do pacote. O supercomputador com pacotes desejados pela comunidade com mais de 64 KB. Quando um supercomputador é iniciado transferência, isso realmente significa negócios e não quer ser interrompido a cada 64 KB. O argumento contra pacotes grandes é que se um pacote de 1 MB atingir um pacote de 1,5 Mbps

Linha T1, esse pacote amarrará a linha por mais de 5 segundos, produzindo uma atraso perceptível para usuários interativos que compartilham a linha. Um compromisso foi alcançado

aqui: pacotes normais são limitados a 64 KB, mas o cabeçalho de extensão salto a salto pode ser usado para permitir jumbogramas.

Um terceiro tópico importante foi a remoção da soma de verificação IPv4. Algumas pessoas compararam isso

mova para remover os freios de um carro. Isso torna o carro mais leve para que ele possa vá mais rápido, mas se um evento inesperado acontecer, você tem um problema.

O argumento contra somas de verificação era que qualquer aplicativo que realmente se importasse sobre a integridade dos dados tem que ter uma soma de verificação da camada de transporte de qualquer maneira, então ter um

outro em IP (além da soma de verificação da camada de enlace) é um exagero. Pele- além disso, a experiência mostrou que calcular a soma de verificação de IP era um importante despesa em IPv4. O acampamento antichecksum venceu este, e o IPv6 não tem um soma de verificação.

Os hosts móveis também foram um ponto de discórdia. Se um computador portátil voar do outro lado do mundo, ele pode continuar operando lá com o mesmo anúncio IPv6 vestido, ou tem que usar esquema com agentes domésticos? Algumas pessoas queriam construir suporte explícito para hosts móveis em IPv6. Esse esforço falhou quando não houve sensus poderia ser encontrado para qualquer proposta específica.

Provavelmente, a maior batalha foi sobre segurança. Todos concordaram que era essencial tial. A guerra era sobre onde colocá-lo e como. Primeiro onde. O argumento para

colocá-lo na camada de rede é que ele se torna um serviço padrão que todos os aplicativos podem ser usados sem qualquer planejamento prévio. O argumento contra isso é que aplicativos realmente seguros geralmente não querem nada menos do que fim a fim criptografia, onde o aplicativo de origem faz a criptografia e o aplicativo de destino implementações de camada de rede com erros sobre as quais ele não tem controle. A resposta a este argumento é que esses aplicativos podem simplesmente abster-se de usar o IP securitários e fazer o trabalho por conta própria. A réplica é que as pessoas que não confie na rede para fazer isso da maneira certa não querem pagar o preço do lento, volumoso Implementações de IP que possuem esse recurso, mesmo se estiver desativado. Outro aspecto de onde colocar a segurança está relacionado ao fato de que muitos (mas não todos) os países têm leis de exportação muito rigorosas em relação à criptografia. Alguns, notavelmente a França e o Iraque, também restringem seu uso no mercado interno, de modo que as pessoas não podem tem segredos do governo. Como resultado, qualquer implementação de IP que usasse um

Página 489

SEC. 5,6

A CAMADA DE REDE NA INTERNET

465

sistema criptográfico forte o suficiente para ser de muito valor não poderia ser exportado dos Estados Unidos (e muitos outros países) para clientes em todo o mundo. Havendo dois conjuntos de software, um para uso doméstico e outro para exportação, é algo que a maioria dos fornecedores de computador se opõe vigorosamente.

Um ponto em que não houve polêmica é que ninguém espera o IPv4 Internet deve ser desligada em uma noite de domingo e voltar como um IPv6 Internet segunda de manhã. Em vez disso, "ilhas" isoladas de IPv6 serão convertidas, em comunicando-se por túneis, como mostramos na Seç. 5.5.3. Como as ilhas IPv6 crescer, eles se fundirão em ilhas maiores. Eventualmente, todas as ilhas irão se fundir, e a Internet será totalmente convertida.

Pelo menos esse era o plano. A implantação provou ser o calcanhar de Aquiles do IPv6. Ele permanece pouco usado, embora todos os principais sistemas operacionais o suportem totalmente.

A maioria das implantações são novas situações em que uma operadora de rede - por exemplo, uma operadora de telefonia móvel - precisa de um grande número de endereços IP. Muitas estratégias

foram definidos para ajudar a facilitar a transição. Entre eles estão maneiras de automatizar configurar icamente os túneis que transportam IPv6 pela Internet IPv4 e as formas de hosts para localizar automaticamente os pontos finais do túnel. Os hosts de pilha dupla têm um IPv4 e uma implementação IPv6 para que eles possam selecionar qual protocolo usar no destino do pacote. Essas estratégias irão agilizar a implantação inicial que parece inevitável quando os endereços IPv4 se esgotam. Para mais informações sobre IPv6, consulte Davies (2008).

5.6.4 Protocolos de Controle da Internet

Além do IP, que é usado para transferência de dados, a Internet tem vários protocolos de controle panion que são usados na camada de rede. Eles incluem ICMP, ARP e DHCP. Nesta seção, veremos cada um deles, por sua vez, descrevendo as versões que correspondem ao IPv4 porque são os protocolos que estão em uso comum. ICMP e DHCP têm versões semelhantes para IPv6; o equivalente a O ARP é chamado de NDP (Neighbour Discovery Protocol) para IPv6.

ICMP - o protocolo de mensagens de controle da Internet

A operação da Internet é monitorada de perto pelos roteadores. Quando algum algo inesperado ocorre durante o processamento de pacotes em um roteador, o evento é relatado ao remetente pelo **ICMP (Internet Control Message Protocol)**. ICMP também é usado para testar a Internet. Cerca de uma dúzia de tipos de mensagens ICMP são definidos. Cada tipo de mensagem ICMP é transportado encapsulado em um pacote IP. O mais importantes estão listados na Figura 5.60.

A mensagem DESTINATION UNREACHABLE é usada quando o roteador não pode localizar o destino ou quando um pacote com o bit *DF* não puder ser entregue porque uma rede de "pacotes pequenos" está no caminho.

466

A CAMADA DE REDE

INDIVÍDUO. 5

Tipo de mensagem

Descrição

Destino inalcançável

O pacote não pôde ser entregue

Tempo excedido

Tempo para viver o campo atingiu 0

Problema de parâmetro

Campo de cabeçalho inválido

Têmpera da fonte

Pacote de choke

Redirecionar

Ensine geografia a um roteador

Resposta de eco e eco

Verifique se uma máquina está viva

Solicitação / resposta de carimbo de data / hora

Igual ao Echo, mas com carimbo de data / hora

Anúncio / solicitação de roteador

Encontre um roteador próximo

Figura 5-60. Os principais tipos de mensagem ICMP.

A mensagem TIME EXCEEDED é enviada quando um pacote é descartado porque seu *O contador TtL (tempo de vida)* atingiu zero. Este evento é um sintoma de que os pacotes estão em loop, ou que os valores do contador estão sendo definidos muito baixos.

Um uso inteligente dessa mensagem de erro é o utilitário **traceroute** que foi desenvolvido operado por Van Jacobson em 1987. O Traceroute encontra os roteadores ao longo do caminho de o host para um endereço IP de destino. Ele encontra essas informações sem qualquer tipo de suporte de rede privilegiado. O método é simplesmente enviar uma sequência de pacotes para o destino, primeiro com um TtL de 1, em seguida, um TtL de 2, 3 e assim por diante. Os contadores

nesses pacotes chegará a zero em roteadores sucessivos ao longo do caminho. Esta rota- Cada um deles enviará obedientemente uma mensagem TIME EXCEEDED de volta ao host. De essas mensagens, o host pode determinar os endereços IP dos roteadores ao longo do caminho, bem como manter estatísticas e tempos em partes do caminho. Não é o que

A mensagem TIME EXCEEDED foi planejada, mas talvez seja a rede mais útil ferramenta de depuração de trabalho de todos os tempos.

A mensagem PARAMETER PROBLEM indica que um valor ilegal foi detectado em um campo de cabeçalho. Este problema indica um bug no IP do host de envio software ou possivelmente no software de um roteador transitado.

A mensagem SOURCE QUENCH foi usada há muito tempo para controlar hosts que eram enviando muitos pacotes. Quando um host recebeu esta mensagem, esperava-se que desacelere. Raramente é usado mais porque quando ocorre congestionamento, estes os pacotes tendem a adicionar mais combustível ao fogo e não está claro como responder a eles.

O controle de congestionamento na Internet agora é feito em grande parte pela ação no trânsito camada de esporte, usando perdas de pacotes como um sinal de congestionamento; vamos estudá-lo em detalhes em

Indivíduo. 6

A mensagem REDIRECT é usada quando um roteador percebe que um pacote parece ser roteado incorretamente. É usado pelo roteador para informar ao host de envio para atualizar para um melhor rota.

As mensagens ECHO e ECHO REPLY são enviadas pelos hosts para ver se um determinado destino é alcançável e atualmente vivo. Ao receber a mensagem ECHO ,

SEC. 5,6

A CAMADA DE REDE NA INTERNET

467

espera-se que o destino envie de volta uma mensagem ECHO REPLY . Estas mensagens são usados no utilitário **ping** que verifica se um host está ativo e na Internet. As mensagens TIMESTAMP REQUEST e TIMESTAMP REPLY são semelhantes, exceto que a hora de chegada da mensagem e a hora de partida da resposta são registado na resposta. Este recurso pode ser usado para medir o desempenho da rede. As mensagens ROUTER ADVERTISEMENT e ROUTER SOLICITATION são usado para permitir que hosts localizem roteadores próximos. Um host precisa saber o endereço IP de em pelo menos um roteador para poder enviar pacotes para fora da rede local. Além dessas mensagens, outras foram definidas. A lista online é agora mantido em www.iana.org/assignments/icmp-parameters .

ARP - O Protocolo de Resolução de Endereço

Embora cada máquina na Internet tenha um ou mais endereços IP, estes endereços não são suficientes para o envio de pacotes. NICs de camada de enlace de dados (rede Placas de interface) como placas Ethernet não entendem endereços de Internet. No caso da Ethernet, cada NIC já fabricada vem equipada com um exclusivo Endereço Ethernet de 48 bits. Os fabricantes de placas de rede Ethernet solicitam um bloco de Endereços Ethernet do IEEE para garantir que não haja duas NICs com o mesmo endereço (para evitar conflitos caso as duas NICs apareçam na mesma LAN). Os NICs enviar e receber quadros com base em endereços Ethernet de 48 bits. Eles não sabem nada em tudo cerca de endereços IP de 32 bits.

Agora surge a questão: como os endereços IP são mapeados na camada de enlace de dados endereços, como Ethernet? Para explicar como isso funciona, vamos usar o exemplo de Figura 5-61, na qual uma pequena universidade com duas redes / 24 é ilustrada. 1 rede (CS) é uma Ethernet comutada no Departamento de Ciência da Computação. prefixo 192.32.65.0/24. A outra LAN (EE), também comutada Ethernet, está em Electrical Engineering e tem o prefixo 192.32.63.0/24. As duas LANs estão conectadas por um roteador IP. Cada máquina em uma Ethernet e cada interface no roteador tem um endereço Ethernet exclusivo, rotulado de *E1* a *E6*, e um endereço IP exclusivo no Rede CS ou EE.

Vamos começar vendo como um usuário no host 1 envia um pacote para um usuário no host 2 na rede CS. Vamos supor que o remetente saiba o nome do destinatário receptor, possivelmente algo como *eagle.cs.uni.edu* . O primeiro passo é encontrar o IP endereço para o host 2. Esta pesquisa é realizada pelo DNS, que estudaremos em Indivíduo. 7. No momento, vamos apenas assumir que o DNS retorna o endereço IP para host 2 (192.32.65.5).

O software da camada superior no host 1 agora constrói um pacote com 192.32.65.5 em o campo de *endereço de destino* e o fornece ao software IP para transmissão. O IP o software pode ver o endereço e ver se o destino está na rede CS, (ou seja, sua própria rede). No entanto, ainda precisa de alguma maneira de encontrar o destino Endereço Ethernet para enviar o quadro. Uma solução é ter um arquivo de configuração em algum lugar no sistema que mapeia endereços IP em endereços Ethernet. Enquanto

468

A CAMADA DE REDE

INDIVÍDUO. 5

Ethernet
interruptor

E3

Rede CS
192.32.65.0/24
IP1 = 192.32.65.7
E2
E5
E1

```

E4
E6
192.32.65.1
IP2 = 192.32.65.5
192.32.63.1
IP4 = 192.32.63.8
IP3 = 192.32.63.3
EE Network
192.32.63.0/24
Roteador
Host 1
Host 2
Host 3
Host 4
Quadro, Armação
Fonte
IP
Fonte
Eth.
Destino
IP
Destino
Eth.
Host 1 a 2, na rede CS
IP1
E1
IP2
E2
Host 1 a 4, na rede CS
IP1
E1
IP4
E3
Host 1 a 4, na rede EE
IP1
E4
IP4
E6

```

Figura 5-61. Duas LANs Ethernet comutadas unidas por um roteador.

esta solução é certamente possível, para organizações com milhares de máquinas manter todos esses arquivos atualizados é um trabalho demorado e sujeito a erros. Uma solução melhor é o host 1 enviar um pacote de transmissão para a Ethernet perguntando a quem pertence o endereço IP 192.32.65.5. A transmissão chegará a cada machine na Ethernet CS, e cada um verificará seu endereço IP. Host 2 sozinho irá responder com seu endereço Ethernet (E2). Desta forma, o host 1 aprende que o anúncio de IP dress 192.32.65.5 está no host com endereço Ethernet E2 . O protocolo usado para fazer esta pergunta e obter a resposta é chamado **ARP (Resolução de Endereço Protocolo)**. Quase todas as máquinas na Internet o executam. ARP é definido em RFC 826.

A vantagem de usar ARP em vez de arquivos de configuração é a simplicidade. o gerenciador de sistema não precisa fazer muito, exceto atribuir a cada máquina um anúncio de IP vestir e decidir sobre as máscaras de sub-rede. ARP faz o resto.

Neste ponto, o software IP no host 1 constrói um quadro Ethernet endereçado a E2 , coloca o pacote IP (endereçado a 192.32.65.5) no campo de carga útil e despeja na Ethernet. Os endereços IP e Ethernet deste pacote são fornecidos em

Fig. 5-61. O Ethernet NIC do host 2 detecta este quadro, reconhece-o como um quadro para si mesmo, recolhe-o e causa uma interrupção. O driver Ethernet extrai o IP pacote da carga útil e o passa para o software IP, que vê que está correto corretamente endereçada e processada.

Várias otimizações são possíveis para fazer o ARP funcionar com mais eficiência. Para começar com, uma vez que uma máquina executa o ARP, ela armazena o resultado em cache, caso precise entre em contato com a mesma máquina em breve. Na próxima vez, ele encontrará o mapeamento em seu próprio cache, eliminando assim a necessidade de uma segunda transmissão. Em muitos casos, o host 2

precisará enviar de volta uma resposta, forçando-a também a executar o ARP para determinar o envio

endereço Ethernet de er. Esta transmissão ARP pode ser evitada tendo o host 1 clude seu mapeamento IP-para-Ethernet no pacote ARP. Quando a transmissão ARP arrives no host 2, o par (192.32.65.7, E1) é inserido no cache ARP do host 2. Na verdade, todas as máquinas na Ethernet podem inserir esse mapeamento em seus caches ARP. Para permitir que os mapeamentos mudem, por exemplo, quando um host é configurado para usar um

novo endereço IP (mas mantém o endereço Ethernet antigo), entradas no cache ARP deve expirar após alguns minutos. Uma maneira inteligente de ajudar a manter as informações em cache

atual e otimizar o desempenho é ter cada máquina transmitindo seu mapeamento quando configurado. Essa transmissão geralmente é feita na forma de um ARP procurando seu próprio endereço IP. Não deve haver uma resposta, mas um lado O efeito da transmissão é criar ou atualizar uma entrada no cache ARP de todos.

Isso é conhecido como **ARP gratuito**. Se uma resposta chegar (inesperadamente), duas máquinas foram atribuídas ao mesmo endereço IP. O erro deve ser resolvido ed pelo gerente de rede antes que ambas as máquinas possam usar a rede.

Agora, vejamos a Figura 5-61 novamente, mas desta vez suponha que o host 1 deseja envie um pacote para o host 4 (192.32.63.8) na rede EE. O Host 1 verá que o o endereço IP de destino não está na rede CS. Ele sabe enviar todos esses off-net- tráfego de trabalho para o roteador, também conhecido como **gateway padrão**. Por con atenção, o gateway padrão é o endereço mais baixo na rede (198.31.65.1).

Para enviar um quadro ao roteador, o host 1 ainda deve saber o endereço Ethernet do interface do roteador na rede CS. Ele descobre isso enviando uma transmissão ARP para 198.31.65.1, a partir do qual aprende E3. Em seguida, ele envia o quadro. O mesmo mecanismos de pesquisa são usados para enviar um pacote de um roteador para o próximo por um sequência de roteadores em um caminho da Internet.

Quando a placa de rede Ethernet do roteador obtém este quadro, ele entrega o pacote ao Software IP. Ele sabe pelas máscaras de rede que o pacote deve ser enviado para a rede EE onde alcançará o host 4. Se o roteador não conhecer o Ether- endereço de rede para o host 4, ele usará o ARP novamente. A tabela da Fig. 5-61 lista os endereços Ethernet e IP de origem e destino que estão presentes nos quadros como observados nas redes CS e EE. Observe que os endereços Ethernet mudam com o quadro em cada rede enquanto os endereços IP permanecem constantes (porque eles indicam os terminais em todas as redes interconectadas).

Também é possível enviar um pacote do host 1 para o host 4 sem o conhecimento do host 1 que o host 4 está em uma rede diferente. A solução é fazer com que o roteador responda ARP's na rede CS para o host 4 e fornecer seu endereço Ethernet, E3, como o re- sponse. Não é possível que o host 4 responda diretamente porque ele não verá o Solicitação ARP (pois os roteadores não encaminham broadcasts de nível Ethernet). O roteador irá em seguida, recebe os quadros enviados para 192.32.63.8 e os encaminha para a rede EE. Essa solução é chamada de **proxy ARP**. É usado em casos especiais em que um host deseja aparecer em uma rede, embora na verdade resida em outra rede. Uma situação comum, por exemplo, é um computador móvel que deseja algum outro nó para coletar pacotes para ele quando não estiver em sua rede doméstica.

DHCP - o protocolo de configuração dinâmica de hosts

ARP (bem como outros protocolos de Internet) assume que os hosts são configurados com algumas informações básicas, como seus próprios endereços IP. Como hosts obtêm essas informações? É possível configurar manualmente cada computador, mas isso é tedioso e sujeito a erros. Existe uma maneira melhor, e é chamada de **DHCP** (**Protocolo de configuração dinâmica de hosts**).

Com o DHCP, toda rede deve ter um servidor DHCP que é responsável por configuração. Quando um computador é iniciado, ele tem uma Ethernet embutida ou outro link endereço de camada embutido na NIC, mas nenhum endereço IP. Muito parecido com ARP, o computador difunde uma solicitação de endereço IP em sua rede. Ele faz isso usando um Pacote DHCP DISCOVER . Este pacote deve chegar ao servidor DHCP. Se aquele servidor não está diretamente conectado à rede, o roteador será configurado para receber O DHCP os transmite e os retransmite ao servidor DHCP, onde quer que esteja. Quando o servidor recebe a solicitação, ele aloca um endereço IP livre e envia para o host em um pacote de OFERTA DHCP (que novamente pode ser retransmitido através do roteador). Para ser capaz de fazer este trabalho mesmo quando os hosts não têm endereços IP, o servidor identifica um host usando seu endereço Ethernet (que é transportado no DHCP Pacote DISCOVER)

Um problema que surge com a atribuição automática de endereços IP de um pool é por quanto tempo um endereço IP deve ser alocado. Se um host sai da rede e não retorna seu endereço IP para o servidor DHCP, esse endereço será permanente definitivamente perdido. Após um período de tempo, muitos endereços podem ser perdidos. Para prevenir isso

aconteça, a atribuição de endereço IP pode ser por um período fixo de tempo, uma tecnologia nique chamado **leasing** . Pouco antes de a concessão expirar, o host deve solicitar um DHCP renovação. Se não conseguir fazer uma solicitação ou se a solicitação for negada, o host não pode usar mais o endereço IP fornecido anteriormente.

O DHCP é descrito nas RFCs 2131 e 2132. É amplamente utilizado na Internet para configurar todos os tipos de parâmetros, além de fornecer hosts com IP vestidos. Bem como em redes empresariais e domésticas, o DHCP é usado pelos ISPs para definir os parâmetros dos dispositivos no link de acesso à Internet, para que os clientes não precisam telefonar para seus ISPs para obter essas informações. Exemplos comuns de informação que está configurada inclui a máscara de rede, o endereço IP do padrão gateway e os endereços IP dos servidores DNS e de horário. DHCP tem amplamente re-colocados protocolos anteriores (chamados RARP e BOOTP) com funções mais limitadas nacionalidade.

5.6.5 Comutação de etiqueta e MPLS

Até agora, em nosso tour pela camada de rede da Internet, focamos exclusivamente em pacotes como datagramas que são encaminhados por roteadores IP. Há sim também outro tipo de tecnologia que está começando a ser amplamente utilizado, especialmente por ISPs, a fim de mover o tráfego da Internet em suas redes. Esta tecnologia é

Página 495

SEC. 5,6

A CAMADA DE REDE NA INTERNET

471

chamado **MPLS (MultiProtocol Label Switching)** e está perigosamente perto de comutação cuit. Apesar de muitas pessoas na comunidade da Internet terem uma aversão intensa por redes orientadas a conexões, a ideia parece manter voltando. Como disse Yogi Berra certa vez, é como um déjà vu novamente. However, existem diferenças essenciais entre a maneira como a Internet lida com a rota de construção e a forma como as redes orientadas à conexão o fazem, então a técnica é certamente não é comutação de circuito tradicional.

MPLS adiciona um rótulo na frente de cada pacote, e o encaminhamento é baseado no rótulo em vez do endereço de destino. Tornando o rótulo um índice em um A tabela final torna a localização da linha de saída correta apenas uma questão de consulta à tabela. Usando essa técnica, o encaminhamento pode ser feito muito rapidamente. Essa vantagem era a motivação original por trás do MPLS, que começou como tecnologia proprietária conhecido por vários nomes, incluindo **troca de tag** . Eventualmente, a IETF começou a padronizar a ideia. Ele é descrito no RFC 3031 e em muitos outros RFCs. Os principais benefícios ao longo do tempo passaram a ser um roteamento flexível e encaminhamento que é adequado para qualidade de serviço, bem como rápido.

A primeira pergunta a fazer é para onde vai o rótulo? Uma vez que os pacotes IP eram

não projetado para circuitos virtuais, não há campo disponível para o número do circuito virtual dentro do cabeçalho IP. Por este motivo, um novo cabeçalho MPLS teve que ser adicionado em frente ao cabeçalho IP. Em uma linha de roteador a roteador usando PPP como protocolo de enquadramento

col, o formato do quadro, incluindo os cabeçalhos PPP, MPLS, IP e TCP, é como mostrado na Fig. 5-62.

PPP
MPLS
IP
Rótulo
QoS S
TtL
20
Bits
Cabeçalhos
3 1
8
TCP
Dados do usuário
CRC

Figura 5-62. Transmitindo um segmento TCP usando IP, MPLS e PPP.

O cabeçalho MPLS genérico tem 4 bytes e quatro campos. Mais importante é o campo *Label*, que contém o índice. O campo *QoS* indica a classe de serviço. O campo *S* está relacionado ao empilhamento de vários rótulos (que é discutido abaixo). O campo *TtL* indica quantas vezes mais o pacote pode ser encaminhado. Isto é decrementado em cada roteador e, se chegar a 0, o pacote será descartado. Este recurso impede o loop infinito no caso de instabilidade de roteamento.

MPLS fica entre o protocolo da camada de rede IP e a camada de link PPP protocol. Não é realmente um protocolo da camada 3 porque depende do IP ou de outra rede

Página 496

472

A CAMADA DE REDE

INDIVÍDUO. 5

endereços de camada para configurar caminhos de rótulo. Não é realmente um protocolo da camada 2 também ser-

faz com que ele encaminhe pacotes em vários saltos, não em um único link. Por esta razão, MPLS às vezes é descrito como um protocolo da camada 2.5. É uma ilustração tão real protocolos nem sempre se encaixam perfeitamente em nosso modelo de protocolo em camadas ideal. No lado bom, porque os cabeçalhos MPLS não fazem parte da rede

pacote de camada ou quadro da camada de enlace de dados, o MPLS é, em grande medida, independente de

ambas as camadas. Entre outras coisas, esta propriedade significa que é possível construir MPLS switches que podem encaminhar pacotes IP e pacotes não IP, dependendo do que Aparece. Este recurso é onde veio o "multiprotocolo" no nome MPLS

de. MPLS também pode transportar pacotes IP em redes não IP.

Quando um pacote MPLS avançado chega em um **LSR (Label Switched Router)**, o rótulo é usado como um índice em uma tabela para determinar a linha de saída a ser usada e também o novo rótulo a ser usado. Esta troca de etiqueta é usada em todas as redes de circuitos virtuais

trabalho. Os rótulos têm apenas significado local e dois roteadores diferentes podem alimentar pacotes relacionados com o mesmo rótulo em outro roteador para transmissão no mesma linha de saída. Para serem distinguíveis na outra extremidade, os rótulos devem ser remapeado a cada salto. Vimos esse mecanismo em ação na Figura 5-3. MPLS usa a mesma técnica.

Como um aparte, algumas pessoas distinguem entre *encaminhar* e *alternar*. Paraguaria é o processo de encontrar a melhor correspondência para um endereço de destino em um para decidir para onde enviar os pacotes. Um exemplo é o prefixo correspondente mais longo algoritmo usado para encaminhamento de IP. Em contraste, a comutação usa um rótulo retirado de o pacote como um índice em uma tabela de encaminhamento. É mais simples e rápido. Estes definições estão longe de ser universais.

Como a maioria dos hosts e roteadores não entendem MPLS, também devemos perguntar

quando e como as etiquetas são anexadas aos pacotes. Isso acontece quando um pacote IP chega ao limite de uma rede MPLS. O **LER** (**Label Edge Router**) inspeciona o endereço IP de destino e outros campos para ver em qual caminho MPLS o pacote deve seguir e coloca a etiqueta correta na frente do pacote. Dentro do Rede MPLS, este rótulo é usado para encaminhar o pacote. Do outro lado do Rede MPLS, o rótulo atendeu ao seu propósito e é removido, revelando o IP pacote novamente para a próxima rede. Esse processo é mostrado na Figura 5-63. Um diferença dos circuitos virtuais tradicionais é o nível de agregação. Com certeza é possível para cada fluxo ter seu próprio conjunto de etiquetas por meio da rede MPLS. No entanto, é mais comum para os roteadores agruparem vários fluxos que terminam em um par- um roteador específico ou LAN e use um único rótulo para eles. Os fluxos que são agrupados juntos sob uma única etiqueta pertencem ao mesmo **FEC** (**Forwarding Classe de equivalência**). Esta aula cobre não apenas para onde os pacotes estão indo, mas também sua classe de serviço (no sentido de serviços diferenciados) porque todo o pacote ets são tratados da mesma maneira para fins de encaminhamento. Com o roteamento de circuito virtual tradicional, não é possível agrupar vários dis- caminhos tintos com diferentes pontos de extremidade no mesmo identificador de circuito virtual porque não haveria como distingui-los no destino final. Com MPLS,

Página 497

SEC. 5,6
A CAMADA DE REDE NA INTERNET

473

Ligando
etiqueta apenas
Mudança de etiqueta
roteador
IP
IP
IP
Rótulo
Borda da etiqueta
roteador
Adicionar
rótulo
Remover
rótulo
(para a próxima
rede)
Rótulo
Rótulo

Figura 5-63. Encaminhando um pacote IP por meio de uma rede MPLS.

os pacotes ainda contêm seu endereço de destino final, além da etiqueta. Em ao final da rota rotulada, o cabeçalho do rótulo pode ser removido e o encaminhamento pode continue da maneira usual, usando o endereço de destino da camada de rede.

Na verdade, o MPLS vai ainda mais longe. Ele pode operar em vários níveis ao mesmo tempo adicionar mais de uma etiqueta à frente de um pacote. Por exemplo, suponha que há muitos pacotes que já têm rótulos diferentes (porque queremos tratar os pacotes de forma diferente em algum lugar da rede) que deve seguir um comp meu caminho para algum destino. Em vez de configurar muitos caminhos de troca de rótulo, um para cada um dos rótulos diferentes, podemos configurar um único caminho. Quando o al- pacotes já etiquetados alcançam o início deste caminho, outro rótulo é adicionado ao frente. Isso é chamado de pilha de rótulos. A etiqueta externa orienta os pacotes pelo caminho. Ele é removido no final do caminho e os rótulos revelados, se houver, são usados para encaminhar o pacote posteriormente. O bit S na Fig. 5-62 permite um roteador removendo um rótulo para saber se há rótulos adicionais restantes. É definido como 1 para o rótulo inferior e 0 para todos os outros rótulos.

A pergunta final que faremos é como as tabelas de encaminhamento de rótulos são configuradas para que os pacotes os seguem. Esta é uma área de grande diferença entre MPLS e projetos convencionais de circuito virtual. Em redes tradicionais de circuito virtual, quando um usuário deseja estabelecer uma conexão, um pacote de configuração é lançado no rede para criar o caminho e fazer as entradas da tabela de encaminhamento. MPLS não

envolver os usuários na fase de configuração. Exigir que os usuários façam qualquer coisa além de enviar um datagrama quebraria muitos softwares de Internet existentes.

Em vez disso, as informações de encaminhamento são configuradas por protocolos que são uma combinação de protocolos de roteamento e protocolos de configuração de conexão. Esses protocolos de controle

são claramente separados do encaminhamento de etiqueta, o que permite múltiplas e diferentes configurações de protocolos tais a serem usados. Uma das variantes funciona assim. Quando um roteador é inicializado, ele verifica para quais rotas é o destino final (por exemplo, quais correções pertencem às suas interfaces). Em seguida, ele cria um ou mais FECs para eles, allocates um rótulo para cada um e passa os rótulos para seus vizinhos. Eles, por sua vez, inserem os rótulos em suas tabelas de encaminhamento e envie novos rótulos para seus vizinhos, até que todos os roteadores tenham adquirido o caminho. Os recursos também podem ser reservados como o

Página 498

474

A CAMADA DE REDE
INDIVÍDUO. 5

caminho é construído para garantir uma qualidade de serviço adequada. Outras variantes podem configurar caminhos diferentes, como caminhos de engenharia de tráfego que ocupam capacidade em conta e criar caminhos sob demanda para oferecer suporte a ofertas de serviços como qualidade de serviço.

Embora as idéias básicas por trás do MPLS sejam diretas, os detalhes são complicado, com muitas variações e casos de uso que estão sendoativamente desenvolvidos. Para obter mais informações, consulte Davie e Farrel (2008) e Davie e Rekhter (2000).

5.6.6 OSPF - Um protocolo de roteamento de gateway interior

Agora concluímos nosso estudo sobre como os pacotes são encaminhados na Internet. É hora de passar para o próximo tópico: roteamento na Internet. Como mencionamos anteriormente, a Internet é composta por um grande número de redes independentes ou ASes (Sistemas Autônomos) que são operados por diferentes organizações, geralmente uma empresa, universidade ou ISP. Dentro de sua própria rede, uma organização pode usar seu próprio algoritmo para roteamento interno, ou **roteamento intradomínio** , pois é mais comumente conhecido. No entanto, há apenas um punhado de protocolos padrão que são populares. Nesta seção, estudaremos o problema de roteamento intradomínio e observe o protocolo OSPF amplamente utilizado na prática. Um roteamento intradomínio protocolo também é chamado de **protocolo de gateway interior** . Na próxima seção, iremos estudar o problema de roteamento entre redes operadas de forma independente, ou **inter-roteamento de domínio** . Para esse caso, todas as redes devem usar a mesma rota entre domínios protocolo de integração ou **protocolo de gateway exterior** . O protocolo que é usado no Internet é BGP (Border Gateway Protocol).

Os primeiros protocolos de roteamento intradomínio usavam um projeto de vetor de distância, com base em

o algoritmo distribuído Bellman-Ford herdado da ARPANET. RIP (Rout- protocolo de informação) é o principal exemplo usado até hoje. Funciona bem em sistemas pequenos, mas não tão bem quanto as redes ficam maiores. Também sofre de problema de contagem ao infinito e convergência geralmente lenta. A ARPANET mudou para um protocolo de estado de link em maio de 1979 por causa desses problemas, e em 1988, a IETF começou a trabalhar em um protocolo de estado de link para roteamento intradomínio. que

protocolo, denominado **OSPF (Open Shortest Path First)**, tornou-se um padrão em 1990. Ele se baseou em um protocolo chamado **IS-IS (Intermediate-System to Intermediate-System)**, que se tornou um padrão ISO. Por causa de sua herança compartilhada, os dois os protocolos são muito mais semelhantes do que diferentes. Para a história completa, consulte RFC 2328. Eles são os protocolos de roteamento intradomínio dominantes e a maioria dos roteadores

dors agora suportam ambos. OSPF é mais amplamente usado em redes de empresas, e IS-IS é mais amplamente usado em redes de ISP. Dos dois, daremos um esboço de como o OSPF funciona.

Dada a longa experiência com outros protocolos de roteamento, o projeto do grupo O OSPF tinha uma longa lista de requisitos que precisavam ser atendidos. Primeiro, o algoritmo tinha a ser publicado na literatura aberta, daí o "O" no OSPF. Um proprietário

Página 499

SEC. 5,6

A CAMADA DE REDE NA INTERNET

475

solução pertencente a uma empresa não serviria. Em segundo lugar, o novo protocolo teve que suportar uma variedade de métricas de distância, incluindo distância física, atraso e assim em. Terceiro, tinha que ser um algoritmo dinâmico, que se adaptasse às mudanças na topologia automática e rapidamente.

Quarto, e novo para o OSPF, ele precisava oferecer suporte ao roteamento com base no tipo de serviço.

O novo protocolo tinha que ser capaz de rotear o tráfego em tempo real de uma maneira e de outra. Isto de uma maneira diferente. Na época, o IP tinha um campo *Tipo de serviço*, mas nenhum protocolo de roteamento usado. Este campo foi incluído no OSPF, mas ainda ninguém o usou, e acabou sendo removido. Talvez esse requisito estivesse à frente de seu tempo, pois precedeu o trabalho da IETF em serviços diferenciados, que rejuvenescerá as aulas de serviço.

Quinto, e relacionado ao acima, OSPF teve que fazer balanceamento de carga, dividindo o carregar em várias linhas. A maioria dos protocolos anteriores enviava todos os pacotes em um único melhor rota, mesmo que houvesse duas rotas igualmente boas. A outra rota não foi usado. Em muitos casos, dividir a carga em várias rotas dá melhor performance.

Em sexto lugar, o suporte para sistemas hierárquicos era necessário. Em 1988, algumas redes tinham crescido tanto que nenhum roteador poderia saber toda a topologia.

O OSPF teve que ser projetado de forma que nenhum roteador o fizesse.

Sétimo, um mínimo de segurança foi necessário para evitar estudiosos que gostam de diversão falhas de roteadores falsificados, enviando-lhes informações de roteamento falsas. Finalmente, provisão era necessária para lidar com roteadores que estavam conectados à Internet através de um túnel. Os protocolos anteriores não lidaram bem com isso.

OSPF suporta links ponto a ponto (por exemplo, SONET) e rede de transmissão funcional (por exemplo, a maioria das LANs). Na verdade, é capaz de suportar redes com vários roteadores, cada um dos quais pode se comunicar diretamente com os outros (chamados **multicamadas sociais**), mesmo que não tenham capacidade de transmissão. Protocolos anteriores não lidaram bem com este caso.

Um exemplo de rede de sistema autônomo é dado na Figura 5.64 (a). Hosts são omitidos porque geralmente não desempenham um papel no OSPF, enquanto os roteadores e redes (que podem conter hosts) sim. A maioria dos roteadores na Figura 5.64 (a) são conectados a outros roteadores por links ponto a ponto, e a redes para alcançar os hosts nessas redes. No entanto, os roteadores R3, R4 e R5 são conectados por um broadcast LAN como Ethernet comutada.

OSPF opera abstraindo a coleção de redes reais, roteadores e links em um gráfico direcionado em que cada arco é atribuído a um peso (distância, atraso, etc.). Uma conexão ponto a ponto entre dois roteadores é representada por um par de arcos, um em cada direção. Seus pesos podem ser diferentes. Uma rede de transmissão é representada por um nó para a própria rede, mas um nó para cada roteador. Os arcos desse nó de rede para os roteadores têm peso 0. Eles são importantes no entanto, sem eles não há caminho pela rede. Outra rede-obra, que têm apenas hospedeiros, têm apenas um arco alcançando-as e não um regirando. Essa estrutura fornece rotas para os hosts, mas não através deles.

476

A CAMADA DE REDE
INDIVÍDUO. 5

LAN 1

LAN 2

LAN 4

LAN 3

R4

R2

R1

R3

R5

R4

R2

R1

R3

R5

LAN 1

LAN 2

LAN 1

LAN 4

(uma)

(b)

0

0

0

3

3

4

5

8

7

5

5

4

4

1

1

1

Figura 5-64. (a) Um sistema autônomo. (b) Uma representação gráfica de (a).

A Figura 5-64 (b) mostra a representação gráfica da rede da Figura 5-64 (a).

O que o OSPF basicamente faz é representar a rede real como um gráfico como isso e, em seguida, use o método de estado do link para que cada roteador calcule o mais curto caminho de si mesmo para todos os outros nós. Vários caminhos podem ser encontrados que são igualmente

curto. Neste caso, o OSPF lembra o conjunto de caminhos mais curtos e durante o pacote encaminhamento, o tráfego é dividido entre eles. Isso ajuda a equilibrar a carga. É chamado

ECMP (Equal Cost MultiPath).

Muitos dos ASes na Internet são grandes e não triviais para o homem era. Para trabalhar nesta escala, o OSPF permite que um AS seja dividido em números numerados **áreas**, onde uma área é uma rede ou um conjunto de redes contíguas. Áreas não se sobrepõem, mas não precisam ser exaustivas, ou seja, alguns roteadores podem não pertencer a nenhuma área.

Os roteadores que ficam totalmente dentro de uma área são chamados de **roteadores internos**. Uma área é um

generalização de uma rede individual. Fora de uma área, seus destinos são visíveis, mas não sua topologia. Essa característica ajuda o roteamento a escalar.

Cada AS tem uma **área de backbone**, chamada área 0. Os roteadores nesta área são chamados de **roteadores de backbone**. Todas as áreas estão conectadas ao backbone, possivelmente por

túneis, então é possível ir de qualquer área do AS para qualquer outra área do AS através do backbone. Um túnel é representado no gráfico como apenas outro arco com um custo. Tal como acontece com outras áreas, a topologia do backbone não é visível fora do espinha dorsal.

Cada roteador conectado a duas ou mais áreas é chamado de **fronteira de área roteador**. Também deve fazer parte do backbone. O trabalho de um roteador de fronteira de área é para resumir os destinos em uma área e injetar este resumo no outro

SEC. 5,6

A CAMADA DE REDE NA INTERNET

477

áreas às quais está conectado. Este resumo inclui informações de custo, mas não todos os detalhes da topologia dentro de uma área. Passar informações de custo permite que os hosts em outras áreas para encontrar o melhor roteador de borda de área para usar para entrar em uma área. Não passando

as informações de topologia reduzem o tráfego e simplificam os cálculos do caminho mais curto de roteadores em outras áreas. No entanto, se houver apenas um roteador de borda fora de um área, mesmo o resumo não precisa ser passado. Rotas para destinos fora de a área sempre começa com a instrução "Vá para o roteador de borda." Este tipo de área é chamada de **área de stub**.

O último tipo de roteador é o **roteador de limite AS**. Ele injeta rotas para exter-destinos finais em outros ASes na área. As rotas externas aparecem como destinos que podem ser alcançados por meio do roteador de limite AS com algum custo. A a rota externa pode ser injetada em um ou mais roteadores de limite AS. A relação-

navio entre ASes, áreas e os vários tipos de roteadores é mostrado na Figura 5.65.

Um roteador pode desempenhar várias funções, por exemplo, um roteador de borda também é um back

roteador de osso.

Área 0 (backbone)

Área 1

Área 2 (esboço)

Espinha dorsal

roteador

Límite AS

roteador

interno

roteador

Límite de área

roteador

1

Autônomo

sistema

Figura 5-65. A relação entre ASes, backbones e áreas no OSPF.

Durante a operação normal, cada roteador dentro de uma área tem o mesmo estado de link banco de dados e executa o mesmo algoritmo de caminho mais curto. Sua principal tarefa é calcular o caminho mais curto de si mesmo para todos os outros roteadores e redes em todo o AS.

Um roteador de fronteira de área precisa dos bancos de dados para todas as áreas às quais está conectado

e deve executar o algoritmo de caminho mais curto para cada área separadamente.

Para uma origem e destino na mesma área, a melhor rota intra-área (que encontra-se totalmente dentro da área) é escolhido. Para uma origem e destino em diferentes áreas, a rota entre áreas deve ir da fonte ao backbone, através do backbone para a área de destino e, em seguida, para o destino. Este algoritmo força uma configuração em estrela no OSPF, com o backbone sendo o hub e o outras áreas sendo raios. Como a rota com o menor custo é escolhida, a rota ers em diferentes partes da rede podem usar diferentes roteadores de fronteira de área para entrar o backbone e a área de destino. Os pacotes são encaminhados da origem ao destino " como estão. " Eles não são encapsulados ou encapsulados (a menos que estejam indo para uma área cuja

478

A CAMADA DE REDE

INDIVÍDUO. 5

apenas a conexão com o backbone é um túnel). Além disso, as rotas para destinos externos pode incluir o custo externo do roteador de limite AS sobre o caminho externo, se desejar, ou apenas o custo interno ao AS.

Quando um roteador é inicializado, ele envia mensagens HELLO em todos os seus pontos a ponto

as linhas e multicast em LANs para o grupo que consiste em todos os outros roteadores.

A partir das respostas, cada roteador descobre quem são seus vizinhos. Roteadores no mesmo LAN são todos vizinhos.

OSPF funciona trocando informações entre roteadores adjacentes, que é não é o mesmo que entre roteadores vizinhos. Em particular, é ineficiente ter cada roteador em uma LAN se comunica com todos os outros roteadores na LAN. Para evitar esta situação

ção, um roteador é eleito como o **roteador designado**. Diz-se que é **adjacente** a todos os outros roteadores em sua LAN e troca informações com eles. Com efeito, ele está atuando como o único nó que representa a LAN. Roteadores vizinhos que não são adjacentes não trocam informações entre si. Um backup de roteador assinado é sempre mantido atualizado para facilitar a transição caso o roteador principal o roteador designado travou e precisa ser substituído imediatamente.

Durante a operação normal, cada roteador inunda periodicamente o LINK STATE ATUALIZE mensagens para cada um de seus roteadores adjacentes. Essas mensagens dão seu estado e fornecer os custos usados no banco de dados topológico. As mensagens de inundação são reconhecido, para torná-los confiáveis. Cada mensagem tem um número de sequência, então um roteador pode ver se um LINK STATE UPDATE de entrada é mais antigo ou mais recente que o que tem atualmente. Os roteadores também enviam essas mensagens quando um link sobe ou para baixo ou seu custo muda.

As mensagens de DESCRIÇÃO DA BASE DE DADOS fornecem os números de sequência de todos os entradas de estado de link atualmente mantidas pelo remetente. Comparando seus próprios valores com

aqueles do remetente, o receptor pode determinar quem tem os valores mais recentes.

Essas mensagens são usadas quando um link é apresentado.

Qualquer um dos parceiros pode solicitar informações sobre o estado do link do outro usando Mensagens de LINK STATE REQUEST. O resultado deste algoritmo é que cada par de roteadores adjacentes verificam quem possui os dados mais recentes e as novas informações está espalhado por toda a área desta forma. Todas essas mensagens são enviadas diretamente no IP pacotes. Os cinco tipos de mensagens estão resumidos na Figura 5.66.

Tipo de mensagem

Descrição

Olá

Usado para descobrir quem são os vizinhos

Atualização do estado do link

Fornece os custos do remetente para seus vizinhos

Link state ack

Reconhece a atualização do estado do link

Descrição do banco de dados

Anuncia quais atualizações o remetente tem

Pedido de estado de link

Solicita informações do parceiro

Figura 5-66. Os cinco tipos de mensagens OSPF.

Página 503

SEC. 5,6

A CAMADA DE REDE NA INTERNET

479

Finalmente, podemos colocar todas as peças juntas. Usando inundação, cada roteador informa todos os outros roteadores em sua área de seus links para outros roteadores e redes e o custo desses links. Esta informação permite que cada roteador construa o gráfico para sua (s) área (s) e calcular os caminhos mais curtos. A área de backbone faz isso trabalho também. Além disso, os roteadores de backbone aceitam informações da área roteadores de fronteira para calcular a melhor rota de cada roteador de backbone para todos os outros roteadores. Esta informação é propagada de volta para os roteadores de fronteira de área,

que o anunciam em suas áreas. Usando essas informações, os roteadores internos podem selecionar a melhor rota para um destino fora de sua área, incluindo a melhor saída roteador para o backbone.

5.6.7 BGP - O Protocolo de Roteamento de Gateway Exterior

Dentro de um único AS, OSPF e IS-IS são os protocolos que são comumente usava. Entre ASes, um protocolo diferente, chamado **BGP (Border Gateway Protocol)**, é usado. Um protocolo diferente é necessário porque os objetivos de um intradomínio protocolo e um protocolo interdomínio não são os mesmos. Tudo um proto intradomínio col tem que fazer é mover os pacotes da forma mais eficiente possível da fonte para o destino inação. Não precisa se preocupar com política.

Em contraste, os protocolos de roteamento entre domínios precisam se preocupar muito com a política

negócio (Metz, 2001). Por exemplo, um AS corporativo pode querer a capacidade de enviar pacotes para qualquer site da Internet e receba pacotes de qualquer site da Internet. Contudo, ele pode não estar disposto a transportar pacotes de trânsito originados em um AS estrangeiro e em um AS estrangeiro diferente, mesmo que seu próprio AS esteja no caminho mais curto entre os dois SAs estrangeiros (" Isso é problema deles, não nosso "). Por outro lado, pode estar disposto a transportar tráfego de trânsito para seus vizinhos, ou mesmo para outros ASes que pagaram por este serviço. As companhias telefônicas, por exemplo, podem ser feliz em atuar como transportadora para seus clientes, mas não para outros. Portal externo protocolos em geral, e BGP em particular, foram projetados para permitir muitos tipos de políticas de roteamento a serem aplicadas no tráfego interAS.

As políticas típicas envolvem considerações políticas, de segurança ou econômicas. UMA alguns exemplos de possíveis restrições de roteamento são:

1. Não transporte tráfego comercial na rede educacional.
2. Nunca envie tráfego do Pentágono em uma rota através do Iraque.
3. Use TeliaSonera em vez da Verizon porque é mais barato.
4. Não use a AT&T na Austrália porque o desempenho é ruim.
5. O tráfego que começa ou termina na Apple não deve transitar pelo Google.

Como você pode imaginar a partir desta lista, as políticas de roteamento podem ser altamente individuais.

Freqüentemente, eles são proprietários porque contêm informações comerciais confidenciais.

Página 504

480

A CAMADA DE REDE INDIVÍDUO. 5

No entanto, podemos descrever alguns padrões que capturam o raciocínio da acima e que são freqüentemente usados como ponto de partida.

Uma política de roteamento é implementada ao decidir qual tráfego pode fluir sobre qual das ligações entre ASes. Uma política comum é que um cliente ISP pague outro provedor ISP para entregar pacotes para qualquer outro destino na Internet e re-pacotes ceive enviados de qualquer outro destino. Diz-se que o ISP do cliente compra **serviço de trânsito** do provedor ISP. É como um cliente em casa comprando Serviço de acesso à Internet de um ISP. Para fazer funcionar, o provedor deve advertir tise rotas para todos os destinos na Internet para o cliente através do link que conecta-os. Desta forma, o cliente terá uma rota para usar para enviar pacotes qualquer lugar. Por outro lado, o cliente deve anunciar rotas apenas para o destino em sua rede para o provedor. Isso permitirá que o provedor envie tráfego para o cliente apenas para esses endereços; o cliente não quer lidar com o tráfego em tendido para outros destinos.

Podemos ver um exemplo de serviço de trânsito na Figura 5.67. Existem quatro ASes que estão conectados. A conexão costuma ser feita com um link em **IXPs (Internet eXchange Points)**, instalações para as quais muitos ISPs têm um link com a finalidade de conectar com outros ISPs. *AS2* , *AS3* e *AS4* são clientes do *AS1* . Eles compram serviço de trânsito a partir dele. Assim, quando a fonte *A* envia para o destino *C* , os pacotes viajar de *AS2* para *AS1* e finalmente para *AS4* . Os anúncios de roteamento viajam em a direção oposta aos pacotes. *AS4* anuncia *C* como um destino para seu trans-sit provider, *AS1* , para permitir que as fontes cheguem a *C* via *AS1* . Mais tarde, *AS1* anuncia uma rota para

C para seus outros clientes, incluindo *AS2* , para que os clientes saibam que eles podem enviar tráfego para *C* via *AS1* .

```
TR  
AS1  
AS2  
AS3  
AS4  
UMA  
EDUCAÇÃO FÍSICA  
CU  
EDUCAÇÃO FÍSICA  
CU  
CU  
TR  
TR  
Caminho de roteamento BGP  
anúncios ( traço )  
Caminho do IP  
pacotes ( sólido )  
Política de roteamento:  
TR = Trânsito  
CU = cliente  
PE = Peer  
B  
C
```

Figura 5-67. Políticas de roteamento entre quatro sistemas autônomos.

Na Figura 5-67, todos os outros ASes compram o serviço de trânsito do *AS1* . Isso fornece com conectividade para que possam interagir com qualquer host na Internet. However, eles têm que pagar por esse privilégio. Suponha que *AS2* e *AS3* trocam muito de tráfego. Dado que suas redes já estão conectadas, se quiserem, eles

Página 505

SEC. 5,6

A CAMADA DE REDE NA INTERNET

481

podem usar uma política diferente - eles podem enviar tráfego diretamente um para o outro gratuitamente.

Isso reduzirá a quantidade de tráfego que o *AS1* deve entregar em seu nome, e esperançosamente reduzirá suas contas. Essa política é chamada de **peering** .

Para implementar o peering, dois ASes enviam anúncios de roteamento um ao outro para os endereços que residem em suas redes. Isso torna possível para *AS2*

para enviar pacotes *AS3* de *A* com destino a *B* e vice-versa. No entanto, observe que o peering não é transitivo. Na Figura 5-67, *AS3* e *AS4* também fazem peering entre si. Isto o peering permite que o tráfego de *C* com destino a *B* seja enviado diretamente para *AS4* . O que aconteceu

canetas se *C* enviar um pacote para *A* ? *AS3* está anunciando apenas uma rota de *B* para *AS4* . Isto é não anunciar uma rota para *um* . A consequência é que o tráfego não passará de *AS4* para *AS3* para *AS2* , embora exista um caminho físico. Esta restrição é exatamente o que o *AS3* deseja. Faz peering com o *AS4* para trocar tráfego, mas não quer transportar tráfego do *AS4* para outras partes da Internet, uma vez que não está sendo pago para isso. No- em vez disso, o *AS4* obtém o serviço de trânsito do *AS1* . Assim, é *AS1* que carregará o pacote de *C* para *A* .

Agora que sabemos sobre trânsito e peering, também podemos ver que *A* , *B* e *C* têm arranjos de trânsito. Por exemplo, *A* deve comprar acesso à Internet da *AS2* . *UMA* pode ser um único computador doméstico ou uma rede corporativa com várias LANs. Como-nunca, ele não precisa executar o BGP porque é uma **rede stub** que está conectada para o resto da Internet por apenas um link. Portanto, o único lugar para enviar pacotes destinado fora da rede está no link para *AS2* . Não há outro lugar para vai. Este caminho pode ser organizado simplesmente configurando uma rota padrão. Por esta razão filho, não mostramos *A* , *B* e *C* como ASes que participam de roteamento entre domínios ing.

Por outro lado, algumas redes de empresas estão conectadas a vários ISPs.

Esta técnica é usada para melhorar a confiabilidade, uma vez que se o caminho através de um ISP falhar, a empresa pode usar o caminho por meio de outro ISP. Esta técnica é chamada **multihoming** . Neste caso, é provável que a rede da empresa execute um interdomínio

protocolo de roteamento (por exemplo, BGP) para informar a outros ASes quais endereços devem ser alcançados através do qual links de ISP.

Muitas variações dessas políticas de trânsito e peering são possíveis, mas elas também pronto para ilustrar como as relações comerciais e o controle sobre a rota do anúncio mentos podem implementar diferentes tipos de políticas. Agora vamos considerar em mais detalhes como os roteadores que executam BGP anunciam rotas entre si e selecionam caminhos pelos quais encaminhar pacotes.

BGP é uma forma de protocolo de vetor de distância, mas é bastante diferente de intradomínio protocolos de vetor de distância, como RIP. Já vimos essa política, em vez de distância mínima, é usado para escolher quais rotas usar. Outra grande diferença é que ao invés de manter apenas o custo da rota para cada destino, cada roteador BGP mantém registro do caminho usado. Esta abordagem é chamada de **caminho vetor protocolo tor**. O caminho consiste no roteador do próximo salto (que pode estar no outro lado do ISP, não adjacente) e a sequência de ASes, ou **caminho AS**, que a rota tem seguido (dado na ordem inversa). Finalmente, pares de roteadores BGP se comunicam

Página 506

482

A CAMADA DE REDE INDIVÍDUO. 5

uns com os outros estabelecendo conexões TCP. Operar desta forma fornece re-comunicação responsável e também esconde todos os detalhes da rede que está sendo passada através.

Um exemplo de como as rotas BGP são anunciadas é mostrado na Figura 5.68. Lá são três ASes e o do meio fornece trânsito para os ISPs esquerdo e direito. UMA O anúncio de rota para o prefixo C começa em AS3. Quando é propagado através do link para R2c no topo da figura, ele tem o caminho AS de simplesmente AS3 e o próximo roteador de salto de R3a. Na parte inferior, ele tem o mesmo caminho AS, mas um próximo salto diferente porque encontrou um link diferente. Este anúncio continua a se propagar e cruza a fronteira em AS1. No roteador R1a, na parte superior da figura, o AS o caminho é AS2, AS3 e o próximo salto é R2a.

R3a
Prefixo
UMA
B
C
AS1
AS2
AS3
Caminho de
pacotes
R3b
R2c
R2d
R2a
R2b
R1a
R1b
C, AS3, R3a
C, AS2, AS3, R2a
C, AS2, AS3, R2b
C, AS2, AS3, R1a
C, AS2, AS3, R1b
Caminho AS
Próximo salto
C, AS3, R3b

Figura 5-68. Propagação de anúncios de rota BGP.

Carregar o caminho completo com a rota torna mais fácil para o recebimento roteador para detectar e interromper loops de roteamento. A regra é que cada roteador que envia um rota fora do AS adiciona seu próprio número de AS à rota. (É por isso que o lista está na ordem inversa.) Quando um roteador recebe uma rota, ele verifica se a sua própria O número AS já está no caminho AS. Se for, um loop foi detectado e o anúncio é descartado. No entanto, e um tanto ironicamente, foi realizado em

no final dos anos 1990 que, apesar dessa precaução, o BGP sofre de uma versão do problema de contagem até o infinito (Labovitz et al., 2001). Não há loops de longa duração, mas às vezes as rotas podem ser lentas para convergir e ter loops transitórios. Fornecer uma lista de ASes é uma maneira muito grosseira de especificar um caminho. Um AS pode ser uma pequena empresa ou uma rede de backbone internacional. Não há como dizer da rota. O BGP nem mesmo tenta porque ASes diferentes podem usar diferentes protocolos intradomínio cujos custos não podem ser comparados. Mesmo se eles pudessem ser comparado, um AS pode não querer revelar suas métricas internas. Este é um dos maneiras pelas quais os protocolos de roteamento entre domínios diferem dos protocolos entre domínios.

Página 507

SEC. 5,6 A CAMADA DE REDE NA INTERNET

483

Até agora, vimos como um anúncio de rota é enviado através do link entre dois ISPs. Ainda precisamos de alguma maneira de propagar as rotas BGP de um lado do ISP para o outro, para que possam ser enviados para o próximo ISP. Esta tarefa pode ser realizada pelo protocolo intradomínio, mas como o BGP é muito bom em escalar para redes grandes funciona, uma variante do BGP é freqüentemente usada. É chamado de **iBGP** (**BGP interno**) para distinguir

adivinhe com o uso regular do BGP como **eBGP** (**BGP externo**).

A regra de propagação de rotas dentro de um ISP é que cada roteador no limite de ISP aprende de todas as rotas vistas por todos os outros roteadores de limite, para consistência. Se um roteador de limite no ISP aprende um prefixo para IP 128.208.0.0/16, todos os outros roteadores aprenderão sobre esse prefixo. O prefixo então será acessível de todas as partes do ISP, não importa como os pacotes entram no ISP a partir de outros ASes.

Não mostramos essa propagação na Figura 5-68 para evitar a desordem, mas, por exemplo, o roteador *R2b* saberá que pode alcançar *C* por meio do roteador *R2c* no topo ou roteador *R2d* na parte inferior. O próximo salto é atualizado conforme a rota cruza dentro do ISP para que os roteadores do outro lado do ISP saibam qual roteador usar para sair do ISP por outro lado. Isso pode ser visto nas rotas mais à esquerda em que o próximo salto aponta para um roteador no mesmo ISP e não para um roteador no próximo ISP.

Agora podemos descrever a peça chave que faltava, que é como os roteadores BGP escolha qual rota usar para cada destino. Cada roteador BGP pode aprender uma rota para um determinado destino do roteador ao qual ele está conectado no próximo ISP e de todos os outros roteadores de limite (que ouviram diferentes rotas do roteadores aos quais estão conectados em outros ISPs). Cada roteador deve decidir qual rota neste conjunto de rotas é o melhor para usar. Em última análise, a resposta é que cabe a o ISP para escrever alguma política para escolher a rota preferida. No entanto, esta explicação é muito geral e nada satisfatória, então podemos pelo menos descrever algumas estratégias comuns.

A primeira estratégia é que as rotas por meio de redes com peering sejam escolhidas de preferência para rotas através de provedores de transporte público. Os primeiros são gratuitos; o último custa dinheiro. Um simi-

Uma estratégia comum é que as rotas dos clientes tenham a preferência mais elevada. É só um bom negócio para enviar tráfego diretamente para os clientes pagantes.

Um tipo diferente de estratégia é a regra padrão de que caminhos AS mais curtos são melhores. Isso é discutível, visto que um AS poderia ser uma rede de qualquer tamanho, então um caminho através de três pequenos ASes poderia ser mais curto do que um caminho através de um grande COMO. No entanto, mais curto tende a ser melhor em média, e essa regra é comum desempatador.

A estratégia final é preferir a rota que possui o menor custo dentro do ISP.

Essa é a estratégia implementada na Figura 5.68. Os pacotes enviados de *A* para *C* saem do *AS1* no roteador superior, *R1a*. Os pacotes enviados de *B* saem pelo roteador inferior, *R1b*. O

razão é que *A* e *B* estão tomando o caminho de custo mais baixo ou o caminho mais rápido para fora *AS1*. Porque eles estão localizados em diferentes partes do ISP, a saída mais rápida para cada um é diferente. A mesma coisa acontece quando os pacotes passam pelo *AS2*. Na última etapa, o *AS3* deve transportar o pacote de *B* por meio de sua própria rede.

Página 508

484

A CAMADA DE REDE
INDIVÍDUO. 5

Essa estratégia é conhecida como **saída antecipada** ou **roteamento batata quente**. Tem o curioso efeito colateral de tender a fazer rotas assimétricas. Por exemplo, considere o caminho tirada quando *C* envia uma volta pacote para *B*. O pacote sairá do *AS3* rapidamente, no roteador superior, para evitar o desperdício de seus recursos. Da mesma forma, ele ficará no topo quando

AS2 passa para *AS1* o mais rápido possível. Então o pacote terá uma longa jornada em *AS1*. Esta é uma imagem de espelho do caminho tomado a partir de *B* para *C*. A discussão acima deve deixar claro que cada roteador BGP escolhe seu próprio melhor rota das possibilidades conhecidas. Não é o caso, como poderia ser ingenuamente expected, que BGP escolhe um caminho a seguir no nível AS e OSPF escolhe caminhos dentro de cada um dos ASes. BGP e o protocolo de gateway interior são integrado muito mais profundamente. Isso significa que, por exemplo, o BGP pode encontrar o melhor ponto de saída de um ISP para o próximo e este ponto irá variar em todo o ISP, como no caso da política da batata quente. Isso também significa que os roteadores BGP em diferentes partes de um AS podem escolher diferentes caminhos do AS para chegar ao mesmo destino. O ISP deve ter cuidado para configurar todos os roteadores BGP para fazer escolhas compatíveis dada toda essa liberdade, mas isso pode ser feito na prática. Surpreendentemente, apenas arranhamos a superfície do BGP. Para mais informações mação, consulte a especificação BGP versão 4 no RFC 4271 e RFCs relacionados. No entanto, perceba que grande parte de sua complexidade está nas políticas, que não são descrito na especificação do protocolo BGP.

5.6.8 Multicast da Internet

A comunicação IP normal é entre um remetente e um receptor. Contudo, para alguns aplicativos, é útil que um processo seja capaz de enviar para um grande número número de receptores simultaneamente. Exemplos são a transmissão de um evento esportivo ao vivo para muitos visualizadores, entregando atualizações do programa para um pool de servidores replicados, e tratamento de chamadas telefônicas de conferência digital (ou seja, com vários participantes). IP suporta comunicação um para muitos, ou multicast, usando classe D IP ad- vestidos. Cada endereço de classe D identifica um grupo de hosts. Vinte e oito bits são disponíveis para a identificação de grupos, então mais de 250 milhões de grupos podem existir no mesmo

Tempo. Quando um processo envia um pacote para um endereço de classe D, uma tentativa de melhor esforço é

feito para entregá-lo a todos os membros do grupo dirigido, mas sem garantias são dados. Alguns membros podem não receber o pacote.

O intervalo de endereços IP 224.0.0.0/24 é reservado para multicast no local rede. Nesse caso, nenhum protocolo de roteamento é necessário. Os pacotes são multicast por simplesmente transmitindo-os na LAN com um endereço multicast. Todos os hosts no LAN recebe as transmissões, e os hosts que são membros do grupo processam o pacote. Os roteadores não encaminham o pacote para fora da LAN. Alguns exemplos de locais endereços multicast são:

224.0.0.1 Todos os sistemas em uma LAN

224.0.0.2 Todos os roteadores em uma LAN

224.0.0.5 Todos os roteadores OSPF em uma LAN

224.0.0.251 Todos os servidores DNS em uma LAN

SEC. 5,6

A CAMADA DE REDE NA INTERNET

485

Outros endereços multicast podem ter membros em redes diferentes. Nisso caso, um protocolo de roteamento é necessário. Mas primeiro os roteadores multicast precisam saber quais hosts são membros de um grupo. Um processo pede a seu host para entrar em um grupo. Ele também pode pedir a seu anfitrião para deixar o grupo. Cada host mantém o controle de quais grupos aos quais seus processos pertencem atualmente. Quando o último processo em um host deixa um grupo, o host não é mais um membro desse grupo. Cerca de uma vez por minuto, cada roteador multicast envia um pacote de consulta para todos os hosts em sua LAN (usando o local endereço multicast de 224.0.0.1, é claro) solicitando que relatem no grupos aos quais eles pertencem atualmente. Os roteadores multicast podem ou não ser colocados com os roteadores padrão. Cada host envia de volta respostas para todos os endereços de classe D em que está interessado. Esses pacotes de consulta e resposta usam um protocolo chamado **IGMP (Internet Group Management Protocol)**. É descrito em RFC 3376.

Qualquer um dos vários protocolos de roteamento multicast pode ser usado para construir multicast árvores abrangentes que fornecem caminhos de remetentes a todos os membros do grupo.

Os algoritmos usados são aqueles que descrevemos na Seç. 5.2.8. Dentro de um AS, o principal protocolo utilizado é o **PIM (Protocol Independent Multicast)**. PIM vem em vários sabores. No modo Denso PIM, um encaminhamento de caminho reverso podado árvore é criada. Isso é adequado para situações em que os membros estão em todos os lugares em a rede, como a distribuição de arquivos para vários servidores em uma rede de data center trabalhos. No modo esparsão do PIM, as árvores de abrangência que são construídas são semelhantes às baseadas em núcleo árvores. Isso é adequado para situações como uma TV multicast de provedor de conteúdo para assinantes em sua rede IP. Uma variante desse design, chamada de fonte específica Multicast PIM, é otimizado para o caso de haver apenas um remetente para o grupo. Finalmente, extensões multicast para BGP ou túneis precisam ser usadas para criar multicast rotas quando os membros do grupo estão em mais de um AS.

5.6.9 IP móvel

Muitos usuários da Internet possuem computadores móveis e desejam permanecer conectados quando eles estão longe de casa e até mesmo na estrada. Infelizmente, o sistema de endereçamento IP torna mais fácil falar do que fazer trabalhar longe de casa, pois iremos descrever em breve. Quando as pessoas começaram a exigir a habilidade de qualquer maneira,

A IETF criou um Grupo de Trabalho para encontrar uma solução. O Grupo de Trabalho rapidamente para mulou uma série de objetivos considerados desejáveis em qualquer solução. Os principais estavam:

1. Cada host móvel deve ser capaz de usar seu endereço IP residencial em qualquer lugar.
2. Mudanças de software nos hosts fixos não eram permitidas.
3. Não foram permitidas alterações no software e nas tabelas do roteador.
4. A maioria dos pacotes para hosts móveis não deve fazer desvios no caminho.
5. Nenhuma sobrecarga deve ocorrer quando um host móvel está em casa.

486

A CAMADA DE REDE

INDIVÍDUO. 5

A solução escolhida foi a descrita na Seç. 5.2.10. Em suma, cada site que deseja permitir que seus usuários façam roaming tem que criar um helper no site chamado de **agente doméstico**. Quando um host móvel aparece em um site estrangeiro, ele obtém um novo IP

endereço (denominado endereço temporário) no site externo. O celular então diz ao agente local onde está agora, fornecendo o endereço de assistência. Quando um pacote para o celular chega ao site inicial e o celular está em outro lugar, o agente doméstico pega o pacote e o encaminha para o celular no endereço atual. O mobile pode enviar pacotes de resposta diretamente para quem está se comunicando, mas ainda usando seu endereço residencial como o endereço de origem. Esta solução atende a todos os re-peculiaridades declaradas acima, exceto que os pacotes para hosts móveis fazem desvios.

Agora que cobrimos a camada de rede da Internet, podemos entrar a solução em mais detalhes. A necessidade de suporte de mobilidade em primeiro lugar vem do próprio esquema de endereçamento IP. Cada endereço IP contém um número de rede e um número de host. Por exemplo, considere a máquina com endereço IP 160.80.40.20/16. O 160.80 fornece o número da rede; o 40.20 é o anfitrião número. Roteadores em todo o mundo têm tabelas de roteamento informando qual link usar para chegar à rede 160.80. Sempre que um pacote chega com um endereço IP de destino do formulário 160.80.xxx.yyy, sai nessa linha. Se de repente, o ma-com esse endereço é transportado para algum site distante, os pacotes para ele continue a ser roteado para sua LAN doméstica (ou roteador).

Nesse estágio, existem duas opções - ambas pouco atraentes. O primeiro é que nós poderia criar uma rota para um prefixo mais específico. Ou seja, se o site distante anunciar uma rota para 160.80.40.20/32, os pacotes enviados para o destino começarão a chegar no lugar certo novamente. Esta opção depende do algoritmo de prefixo de correspondência mais longo que é usado em roteadores. No entanto, adicionamos uma rota a um prefixo IP com um sene-gle o endereço IP nele. Todos os ISPs do mundo aprenderão sobre esse prefixo. Se todos muda as rotas globais de IP desta forma quando movem seu computador, cada roteador teria milhões de entradas na tabela, a um custo astronômico para a Internet. Isto opção não é viável.

A segunda opção é alterar o endereço IP do celular. Verdade, pacotes enviado para o endereço IP residencial não será mais entregue até que todas as pessoas relevantes ple, programas e bancos de dados são informados sobre a mudança. Mas o celular ainda pode use a Internet no novo local para navegar na Web e executar outros aplicativos.

Esta opção lida com a mobilidade em uma camada superior. É o que normalmente acontece quando um o usuário leva um laptop a uma cafeteria e usa a Internet via rede sem fio local rede. A desvantagem é que ele quebra alguns aplicativos, e não mantenha a conectividade enquanto o celular se move.

Como um aparte, a mobilidade também pode ser tratada em uma camada inferior, a camada de link. Isto é o que acontece ao usar um laptop em uma única rede sem fio 802.11. O IP o endereço do celular não muda e o caminho da rede permanece o mesmo. isto é o link sem fio que fornece mobilidade. No entanto, o grau de mobilidade é limitado. Se o laptop for muito longe, ele terá que se conectar à Internet por meio de um outra rede com um endereço IP diferente.

A solução de IP móvel para IPv4 é fornecida no RFC 3344. Funciona com o roteamento de Internet existente e permite que os hosts permaneçam conectados com seu próprio anúncio de IP vestidos à medida que se movem. Para que funcione, o celular deve ser capaz de descobrir quando ele mudou. Isso é realizado com anúncio de roteador ICMP e mensagens de solicitação. Dispositivos móveis ouvem anúncios periódicos de roteadores ou enviam um solicitação para descobrir o roteador mais próximo. Se este roteador não for o endereço normal de o roteador quando o celular está em casa, ele deve estar em uma rede estrangeira. Se este

o roteador mudou desde a última vez, o celular mudou para outra rede estrangeira trabalhos. Esse mesmo mecanismo permite que os hosts móveis encontrem seus agentes domésticos. Para obter um endereço IP seguro na rede estrangeira, um celular pode simplesmente usar DHCP. Como alternativa, se os endereços IPv4 forem escassos, o celular pode enviar e receber pacotes por meio de um agente estrangeiro que já tenha um endereço IP na rede-trabalhos. O host móvel encontra um agente estrangeiro usando o mesmo mecanismo ICMP usado para encontrar o agente doméstico. Depois que o celular obtém um endereço IP ou encontra um fornecedor estrangeiro, é capaz de usar a rede para enviar uma mensagem ao seu agente doméstico, informando o agente local de sua localização atual.

O agente doméstico precisa de uma maneira de interceptar os pacotes enviados apenas para o celular quando o celular não está em casa. O ARP fornece um mecanismo conveniente. Enviar um pacote por Ethernet para um host IP, o roteador precisa saber o anúncio Ethernet vestido do anfitrião. O mecanismo usual é o roteador enviar uma consulta ARP para perguntar, por exemplo, qual é o endereço Ethernet de 160.80.40.20. Quando o celular está em casa, ele responde a consultas ARP para seu endereço IP com seu próprio anúncio Ethernet vestir. Quando o celular está ausente, o agente doméstico responde a esta consulta dando seu endereço Ethernet. O roteador então envia pacotes para 160.80.40.20 para a casa agente. Lembre-se de que isso é chamado de ARP de proxy.

Para atualizar rapidamente os mapeamentos ARP para frente e para trás quando o celular sai casa ou chega em casa, outra técnica ARP chamada **ARP gratuito** pode ser usado. Basicamente, o agente móvel ou doméstico envia a si mesmo uma consulta ARP para o endereço IP móvel que fornece a resposta certa para que o roteador perceba e atualiza seu mapeamento.

Túnel para enviar um pacote entre o agente doméstico e o host móvel no endereço de cuidado é feito encapsulando o pacote com outro cabeçalho de IP destinado para o endereço cuidado. Quando o pacote encapsulado chega aos cuidados de endereço, o cabeçalho IP externo é removido para revelar o pacote.

Tal como acontece com muitos protocolos de Internet, o diabo está nos detalhes e, na maioria das vezes, o

detalhes de compatibilidade com outros protocolos implantados. São dois complicações. Primeiro, as caixas NAT dependem de espreitar além do cabeçalho IP para olhar o cabeçalho TCP ou UDP. A forma original de tunelamento para IP móvel não usava esses cabeçalhos, por isso não funcionou com caixas NAT. A solução foi mudar o encapsulamento para incluir um cabeçalho UDP.

A segunda complicação é que alguns ISPs verificam os endereços IP de origem de pacotes para ver se eles correspondem onde o protocolo de roteamento acredita que a fonte deve ser localizado. Esta técnica é chamada de **filtragem de entrada** e é um instrumento de segurança

Página 512

488

A CAMADA DE REDE INDIVÍDUO. 5

medida destinada a descartar o tráfego com endereços aparentemente incorretos que podem seja malicioso. No entanto, os pacotes enviados do celular para outros hosts da Internet quando está em uma rede estrangeira terá um endereço IP de origem que está fora do lugar, então eles será descartado. Para contornar esse problema, o celular pode usar o anúncio cuidadoso vestir-se como uma fonte para fazer um túnel dos pacotes de volta ao agente doméstico. Daqui, eles são enviados para a Internet do que parece ser o local certo. O custo é que a rota é mais rotunda.

Outro assunto que não discutimos é a segurança. Quando um agente doméstico obtém um mensageiro solicitando que encaminhe todos os pacotes de Roberta para algum endereço IP, é melhor não cumprir, a menos que esteja convencido de que Roberta é a fonte deste rebusca, e não alguém tentando se passar por ela. Autenticação criptográfica

protocolos são usados para este propósito. Estudaremos esses protocolos no Cap. 8

Os protocolos de mobilidade para IPv6 têm como base o IPv4. O esquema acima

sofre do problema de roteamento de triângulo em que os pacotes enviados para o celular levam um cãozinho por meio de um agente doméstico distante. No IPv6, a otimização da rota é usada para seguir

baixo um caminho direto entre o celular e outros endereços IP após o pacote inicial
ets seguiram o longo caminho. O IPv6 móvel é definido no RFC 3775.

Existe outro tipo de mobilidade que também está sendo definido para a Internet.

Alguns aviões têm rede sem fio embutida que os passageiros podem usar para controlar conectar seus laptops à Internet. O avião tem um roteador que se conecta ao resto da Internet por meio de um link sem fio. (Você esperava um link com fio?) Então, agora nós tem um roteador voador, o que significa que toda a rede é móvel. Rede projetos de mobilidade suportam essa situação sem que os laptops percebam que o avião é móvel. Para eles, é apenas mais uma rede. Claro, alguns dos laptops podem estar usando IP móvel para manter seus endereços residenciais enquanto eles estão no avião, então temos dois níveis de mobilidade. Mobilidade de rede é de-multado por IPv6 no RFC 3963.

5.7 RESUMO

A camada de rede fornece serviços para a camada de transporte. Pode ser baseado em datagramas ou circuitos virtuais. Em ambos os casos, sua principal tarefa é o roteamento de pacotes da origem ao destino. Em redes de datagramas, uma decisão de roteamento é feito em cada pacote. Em redes de circuito virtual, é feito quando o circuito virtual

cuit está configurado.

Muitos algoritmos de roteamento são usados em redes de computadores. Inundar é um processo simples

algoritmo para enviar um pacote por todos os caminhos. A maioria dos algoritmos encontra o caminho mais curto

e se adaptar às mudanças na topologia da rede. Os principais algoritmos são distância roteamento de vetor e roteamento de estado de link. A maioria das redes reais usa um desses.

Outros tópicos importantes de roteamento são o uso de hierarquia em grandes redes, roteamento para hosts móveis e roteamento de broadcast, multicast e anycast.

Página 513

SEC. 5,7
RESUMO

489

As redes podem ficar facilmente congestionadas, levando a um maior atraso e perda pacotes. Os designers de rede tentam evitar o congestionamento projetando a rede ter capacidade suficiente, escolhendo rotas não congestionadas, recusando-se a aceitar mais tráfego, fontes de sinalização para desacelerar e redução de carga.

A próxima etapa além de apenas lidar com o congestionamento é realmente tentar alcançar uma qualidade de serviço prometida. Alguns aplicativos se preocupam mais com o rendimento enquanto outros se preocupam mais com atrasos e instabilidade. Os métodos que podem ser usados para

fornecer diferentes qualidades de serviço, incluindo uma combinação de modelagem de tráfego, reserva de recursos em roteadores e controle de admissão. Abordagens que foram projetado para uma boa qualidade de serviço inclui serviços integrados IETF (incluindo RSVP) e serviços diferenciados.

As redes diferem de várias maneiras, então, quando várias redes são interconectado, podem ocorrer problemas. Quando diferentes redes têm diferentes máximos tamanhos de pacotes, pode ser necessária fragmentação. Redes diferentes podem funcionar de forma diferente

protocolos de roteamento internamente, mas precisam executar um protocolo comum externamente. Alguns-

vezes os problemas podem ser resolvidos por um túnel de um pacote através de uma rede hostil-funcionar, mas se as redes de origem e de destino forem diferentes, essa abordagem falhará.

A Internet possui uma grande variedade de protocolos relacionados à camada de rede.

Isso inclui o protocolo de datagrama, IP e protocolos de controle associados, como ICMP, ARP e DHCP. Um protocolo orientado a conexão chamado MPLS transporta IP

pacotes em algumas redes. Um dos principais protocolos de roteamento usados na rede funciona é OSPF e o protocolo de roteamento usado nas redes é BGP. The Internet está ficando sem endereços IP rapidamente, então uma nova versão do IP, IPv6, foi desenvolvida e está sendo implantada muito lentamente.

PROBLEMAS

1. Dê dois exemplos de aplicativos de computador para os quais o serviço orientado à conexão é adequado apropriado. Agora dê dois exemplos de qual serviço sem conexão é o melhor.
2. As redes de datagrama roteiam cada pacote como uma unidade separada, independente de todas as outras. As redes de circuitos virtuais não precisam fazer isso, uma vez que cada pacote de dados segue uma rota encerrada. Essa observação significa que as redes de circuito virtual não precisam a capacidade de rotear pacotes isolados de uma fonte arbitrária para um destino arbitrário ção? Explique sua resposta.
3. Dê três exemplos de parâmetros de protocolo que podem ser negociados quando uma conexão está configurada.
4. Supondo que todos os roteadores e hosts estejam funcionando corretamente e que todo o software em ambos está livre de todos os erros, há alguma chance, por menor que seja, de que um pacote seja entregue para o destino errado?

Página 514

490

A CAMADA DE REDE INDIVÍDUO. 5

5. Dê uma heurística simples para encontrar dois caminhos através de uma rede de uma determinada fonte para um determinado destino que pode sobreviver à perda de qualquer linha de comunicação (assumindo que dois tais caminhos existem). Os roteadores são considerados confiáveis o suficiente, por isso não é necessário se preocupar com a possibilidade de travamento do roteador.
6. Considere a rede da Figura 5.12 (a). O roteamento do vetor de distância é usado, e o seguinte vetores ing acabaram de chegar ao roteador C : de B : (5, 0, 8, 12, 6, 2); de D : (16, 12, 6, 0, 9, 10); e de E : (7, 6, 3, 9, 0, 4). O custo dos links de C para B , D e E , são 6, 3 e 5, respectivamente. Qual é a nova tabela de roteamento de C ? Dê a ambos a saída linha para usar e o custo.
7. Se os custos forem registrados como números de 8 bits em uma rede de 50 roteadores e os vetores de distância forem trocada duas vezes por segundo, quanta largura de banda por linha (full-duplex) é consumida pelo algoritmo de roteamento distribuído? Suponha que cada roteador tenha três linhas para outros roteadores.
8. Na Figura 5-13, o OR booleano dos dois conjuntos de bits ACF é 111 em cada linha. É isto apenas um acidente aqui ou isso vale para todas as redes em todas as circunstâncias?
9. Para roteamento hierárquico com 4800 roteadores, quais regiões e tamanhos de cluster devem ser escolhidos para minimizar o tamanho da tabela de roteamento para uma hierarquia de três camadas? Um bom ponto de partida é a hipótese de que uma solução com k clusters de k regiões de k roteadores está próximo do ótimo, o que significa que k é aproximadamente a raiz cúbica de 4800 (cerca de 16). Use tentativa e erro para verificar as combinações em que todos os três parâmetros estão na vizinhança geral de 16.
10. No texto, afirma-se que, quando um host móvel não está em casa, os pacotes são enviados para o seu LAN doméstica são interceptados por seu agente doméstico nessa LAN. Para uma rede IP em um 802.3 LAN, como o agente doméstico realiza essa interceptação?
11. Olhando para a rede da Figura 5-6, quantos pacotes são gerados por um broadcast de B , usando
 - (a) encaminhamento de caminho reverso?
 - (b) a árvore que afunda?
12. Considere a rede da Figura 5.15 (a). Imagine que uma nova linha é adicionada, entre F e G , mas a árvore sumidouro da Figura 5.15 (b) permanece inalterada. Que mudanças ocorrem em Fig. 5-15 (c)?
13. Calcule uma árvore de abrangência multicast para o roteador C na seguinte rede para um grupo com membros em routers A , B , C , D , E , F , I , e K .

UMA
G
H
Eu
eu
D
K
B
C
F
E

INDIVÍDUO. 5
PROBLEMAS
491

14. Suponha que o nó B na Fig. 5-20 acabou de ser reinicializado e não tem informações de roteamento em suas tabelas. De repente, necessita de uma rota para H . Ele envia transmissões com TTL definido como 1, 2, 3 e assim por diante. Quantas rodadas são necessárias para encontrar uma rota?
15. Como um possível mecanismo de controle de congestionamento em uma rede usando circuitos virtuais interfinalmente, um roteador pode se abster de reconhecer um pacote recebido até que (1) saiba sua última transmissão ao longo do circuito virtual foi recebida com sucesso e (2) tem um buffer livre. Para simplificar, suponha que os roteadores usem um protocolo stop-and-wait e que cada circuito virtual tem um buffer dedicado a ele para cada direção de tráfego. Se isso leva T segundos para transmitir um pacote (dados ou confirmação) e há n roteadores o caminho, qual é a taxa em que os pacotes são entregues ao host de destino? Como suponha que os erros de transmissão sejam raros e que a conexão host-roteador seja infinitamente velozes.
16. Uma rede de datagramas permite que os roteadores descartem pacotes sempre que necessário. A probabilidade de um roteador descartar um pacote é p . Considere o caso de um host de origem conectado ao roteador de origem, que está conectado ao roteador de destino, e então para o host de destino. Se qualquer um dos roteadores descarta um pacote, o host de origem até tempo limite e tente novamente. Se as linhas host-roteador e roteador-roteador forem contadas como lúpulo, qual é o número médio de
- (a) saltos que um pacote faz por transmissão?
 - (b) transmissões que um pacote faz?
 - (c) saltos necessários por pacote recebido?
17. Descreva duas diferenças principais entre o método ECN e o método RED de prevenção de congestionamento.
18. Um esquema de token bucket é usado para modelagem de tráfego. Um novo token é colocado no balde a cada 5 μ seg. Cada token é bom para um pacote curto, que contém 48 bytes de dados. Qual é a taxa de dados máxima sustentável?
19. Um computador em uma rede de 6 Mbps é regulado por um token bucket. O token bucket é preenchido a uma taxa de 1 Mbps. Ele está inicialmente lotado com 8 megabits. Quão mais o computador pode transmitir em 6 Mbps?
20. A rede da Figura 5.34 usa RSVP com árvores multicast para os hosts 1 e 2, conforme mostrado. Suponha que o host 3 solicite um canal de largura de banda de 2 MB / s para um fluxo do host 1 e outro canal de largura de banda de 1 MB / s para um fluxo do host 2. Ao mesmo tempo, o host 4 solicita um canal de largura de banda de 2 MB / s para um fluxo do host 1 e host 5 rebusca um canal de largura de banda de 1 MB / s para um fluxo do host 2. Quanto total a largura de banda será reservada para essas solicitações nos roteadores A, B, C, E, H, J, K e L ?
21. Um roteador pode processar 2 milhões de pacotes / s. A carga oferecida a ele é de 1,5 milhão de embalagens ets / seg em média. Se uma rota da origem ao destino contém 10 roteadores, como muito tempo é gasto sendo enfileirado e atendido pelo roteador?
22. Considere o usuário de serviços diferenciados com encaminhamento rápido. Tem alguma garante que os pacotes expedidos tenham um atraso menor do que os pacotes regulares? Por que ou por que não?

492
A CAMADA DE REDE
INDIVÍDUO. 5

23. Suponha que o host A esteja conectado a um roteador $R1$, $R1$ esteja conectado a outro roteador, $R2$, e $R2$ está ligado ao hospedeiro B . Suponha que uma mensagem TCP que contém 900 bytes de dados e 20 bytes de cabeçalho TCP são passados para o código IP no host A para entrega de B . Mostrar o comprimento total, identificação, DF , MF e campos de deslocamento de fragmento de o cabeçalho IP em cada pacote transmitido pelos três links. Suponha que o link $A-R1$ pode suportar um tamanho máximo de quadro de 1024 bytes, incluindo um cabeçalho de quadro de 14 bytes, o link $R1-R2$ pode suportar um tamanho máximo de quadro de 512 bytes, incluindo um quadro de 8 bytes cabeçalho e link $R2-B$ pode suportar um tamanho máximo de quadro de 512 bytes, incluindo um Cabeçalho de quadro de 12 bytes.
24. Um roteador está emitindo pacotes IP cujo comprimento total (dados mais cabeçalho) é de 1024 bytes. Supondo que os pacotes durem 10 segundos, qual é a velocidade máxima da linha que o roteador pode operar sem perigo de percorrer o espaço do número de ID do datagrama IP?

- 25.** Um datagrama IP usando a opção de *roteamento de origem estrita* deve ser fragmentado. Você acha que a opção é copiada em cada fragmento, ou é suficiente apenas colocá-lo no primeiro fragmento? Explique sua resposta.
- 26.** Suponha que em vez de usar 16 bits para a parte de rede de um endereço de origem de classe B finalmente, 20 bits foram usados. Quantas redes de classe B haveria?
- 27.** Converta o endereço IP cuja representação hexadecimal é C22F1582 para pontilhada notação decimal.
- 28.** Uma rede na Internet tem uma máscara de sub-rede de 255.255.240.0. Qual é o máximo número de hosts que ele pode manipular?
- 29.** Embora os endereços IP sejam tentados para redes específicas, os endereços Ethernet não. Você pode pensar em um bom motivo pelo qual eles não são?
- 30.** Um grande número de endereços IP consecutivos estão disponíveis a partir de 198.16.0.0. Supõem que quatro organizações, A , B , C e D , solicitem 4000, 2000, 4000 e 8000 anúncios vestidos, respectivamente, e nessa ordem. Para cada um deles, dê o primeiro endereço IP como-assinado, o último endereço IP atribuído e a máscara na notação *wxyz / s* .
- 31.** Um roteador acaba de receber os seguintes novos endereços IP: 57.6.96.0/21, 57.6.104.0/21, 57.6.112.0/21 e 57.6.120.0/21. Se todos eles usarem a mesma saída linha, eles podem ser agregados? Se sim, para quê? Se não, porque não?
- 32.** O conjunto de endereços IP de 29.18.0.0 a 19.18.128.255 foi agregado a 29.18.0.0/17. No entanto, há uma lacuna de 1.024 endereços não atribuídos de 29.18.60.0 a 29.18.63.255 que agora são repentinamente atribuídos a um host usando uma saída diferente linha. Agora é necessário dividir o endereço agregado em seus blocos constituintes, adicionar o novo bloco à tabela, e então ver se alguma reagregação é possível? Se não, o que pode ser feito em vez disso?
- 33.** Um roteador tem as seguintes entradas (CIDR) em sua tabela de roteamento:

Endereço / máscara Próximo salto

135.46.56.0/22	Interface 0
135.46.60.0/22	Interface 1
192.53.40.0/23	Roteador 1
padrão	
Roteador 2	

Página 517

INDIVÍDUO. 5

PROBLEMAS

493

Para cada um dos seguintes endereços IP, o que o roteador faz se um pacote com aquele endereço chega?

- (a) 135.46.63.10
- (b) 135.46.57.14
- (c) 135.46.52.2
- (d) 192.53.40.7
- (e) 192.53.56.7

34. Muitas empresas têm uma política de ter dois (ou mais) roteadores conectando o computador para a Internet para fornecer alguma redundância no caso de um deles cair. É esta política ainda é possível com o NAT? Explique sua resposta.

35. Você acabou de explicar o protocolo ARP a um amigo. Quando estiver tudo pronto, ele diz: " Já sei. ARP fornece um serviço para a camada de rede, por isso faz parte dos dados camada de link. " O que você diz a ele?

36. Descreva uma maneira de remontar fragmentos de IP no destino.

37. A maioria dos algoritmos de remontagem de datagramas IP tem um temporizador para evitar a perda de um fragmento

prenda os buffers de remontagem para sempre. Suponha que um datagrama esteja fragmentado em quatro fragmentos. Os primeiros três fragmentos chegam, mas o último está atrasado. Eventualmente, o cronômetro dispara e os três fragmentos da memória do receptor são descartados. UMA pouco depois, o último fragmento entra. O que fazer com ele?

38. No IP, a soma de verificação cobre apenas o cabeçalho e não os dados. Por que você acha isso o projeto foi escolhido?

39. Uma pessoa que mora em Boston viaja para Minneapolis, levando seu computador portátil com ela. Para sua surpresa, a LAN em seu destino em Minneapolis é um IP sem fio LAN, para que ela não precise se conectar. Ainda é necessário percorrer todo o barramento-entrar em contato com agentes locais e estrangeiros para fazer com que o e-mail e outro tráfego cheguem retamente?

40. IPv6 usa endereços de 16 bytes. Se um bloco de 1 milhão de endereços for alocado a cada picosegundo, quanto tempo durarão os endereços?

41. O campo *Protocolo* usado no cabeçalho IPv4 não está presente no cabeçalho IPv6 fixo.

Por que não?

42. Quando o protocolo IPv6 é introduzido, o protocolo ARP precisa ser alterado? E se então, as mudanças são conceituais ou técnicas?
43. Escreva um programa para simular o roteamento usando inundação. Cada pacote deve conter um contador que é diminuído em cada salto. Quando o contador chega a zero, o pacote é descartado. O tempo é discreto, com cada linha tratando de um pacote por intervalo de tempo. Faça três versões do programa: todas as linhas são inundadas, todas as linhas, exceto a entrada as linhas são inundadas e apenas as melhores k linhas (escolhidas estaticamente) são inundadas. Comparar inundação com roteamento determinístico ($k = 1$) em termos de atraso e largura de banda usava.
44. Escreva um programa que simule uma rede de computadores usando o tempo discreto. O primeiro o pacote em cada fila do roteador faz um salto por intervalo de tempo. Cada roteador tem apenas um número finito de buffers. Se um pacote chega e não há espaço para ele, ele é descartado

Página 518

494

A CAMADA DE REDE

INDIVÍDUO. 5

e não retransmitida. Em vez disso, existe um protocolo de ponta a ponta, completo com tempo saídas e pacotes de confirmação, que eventualmente regeneram o pacote do roteador de origem. Trace a taxa de transferência da rede como uma função do tempo de ponta a ponta intervalo de saída, parametrizado pela taxa de erro.

45. Escreva uma função para fazer o encaminhamento em um roteador IP. O procedimento tem um parâmetro, um endereço IP. Ele também tem acesso a uma tabela global que consiste em uma série de triplos. Cada triplô contém três inteiros: um endereço IP, uma máscara de sub-rede e a linha de contorno usar. A função procura o endereço IP na tabela usando CIDR e retorna o

linha para usar como seu valor.

46. Use os programas *traceroute* (UNIX) ou *tracert* (Windows) para rastrear a rota de seu computador para várias universidades em outros continentes. Faça uma lista de transoceânicos links que você descobriu. Alguns sites para experimentar são

www.berkeley.edu (Califórnia)

www.mit.edu (Massachusetts)

www.vu.nl (Amsterdã)

www.ucl.ac.uk (Londres)

www.usyd.edu.au (Sydney)

www.u-tokyo.ac.jp (Tóquio)

www.uct.ac.za (Cidade do Cabo)

Página 519

6

A CAMADA DE TRANSPORTE

Junto com a camada de rede, a camada de transporte é o coração do protocolo hierarquia col. A camada de rede fornece entrega de pacotes ponta a ponta usando dados gramas ou circuitos virtuais. A camada de transporte baseia-se na camada de rede para vide o transporte de dados de um processo em uma máquina de origem para um processo em um destino

máquina de instalação com um nível desejado de confiabilidade que é independente do físico redes atualmente em uso. Ele fornece as abstrações que os aplicativos precisam para usar a rede. Sem a camada de transporte, todo o conceito de protocolo em camadas colis faria pouco sentido. Neste capítulo, estudaremos a camada de transporte em detalhes, incluindo seus serviços e escolha de design de API para lidar com questões de confiabilidade

, conexões e controle de congestionamento, protocolos como TCP e UDP, e performance.

6.1 O SERVIÇO DE TRANSPORTE

Nas seções a seguir, forneceremos uma introdução ao serviço de transporte. Vemos que tipo de serviço é fornecido à camada de aplicativo. Para tornar a questão do serviço de transporte mais concreta, examinaremos dois conjuntos de primitivas da camada de transporte. Primeiro vem um simples (mas hipotético) para mostrar o idéias básicas. Em seguida, vem a interface comumente usada na Internet.

495

Página 520

496

A CAMADA DE TRANSPORTE
INDIVÍDUO. 6

6.1.1 Serviços prestados às camadas superiores

O objetivo final da camada de transporte é fornecer eficiência, confiabilidade e serviço de transmissão de dados de baixo custo para seus usuários, normalmente processos no topo da camada de aplicação. Para conseguir isso, a camada de transporte faz uso dos serviços fornecidos pela camada de rede. O software e / ou hardware dentro do transporte camada que faz o trabalho é chamada de **entidade de transporte**. A entidade de transporte pode estar localizado no kernel do sistema operacional, em um pacote de biblioteca ligado à rede aplicativos, em um processo de usuário separado, ou mesmo na placa de interface de rede. As duas primeiras opções são mais comuns na Internet. A relação (lógica) das camadas de rede, transporte e aplicação é ilustrado na Fig. 6-1.

Aplicação / transporte
interface
Rede de transporte
interface
Inscrição
(ou sessão)
camada
Transporte
entidade
Transporte
endereço
Rede
endereço
Camada de rede
Inscrição
(ou sessão)
camada
Transporte
entidade
Camada de rede
Segmento
Transporte
protocolo
Host 1
Host 2

Figura 6-1. As camadas de rede, transporte e aplicativo.

Assim como existem dois tipos de serviço de rede, orientado para conexão e conexão sem conexão, também existem dois tipos de serviço de transporte. O orientado à conexão serviço de transporte é semelhante ao serviço de rede orientado à conexão em muitas maneiras. Em ambos os casos, as conexões têm três fases: estabelecimento, transferência de dados, e solte. Endereçamento e controle de fluxo também são semelhantes em ambas as camadas. Além disso, o serviço de transporte sem conexão também é muito semelhante ao serviço de rede sem conexão. No entanto, observe que pode ser difícil fornecer um serviço de transporte sem conexão em cima de um serviço de rede orientado para conexão, uma vez que é ineficiente configurar uma conexão para enviar um único pacote e depois rasgá-lo para baixo imediatamente depois.

A questão óbvia é esta: se o serviço da camada de transporte é tão semelhante ao serviço da camada de rede, por que existem duas camadas distintas? Por que uma camada não é

Página 521

SEC. 6,1
O SERVIÇO DE TRANSPORTE
497

adequado? A resposta é sutil, mas crucial. O código de transporte é executado inteiramente em as máquinas dos usuários, mas a camada de rede é executada principalmente nos roteadores, que são operado pela operadora (pelo menos para uma rede de área ampla). O que acontece se o camada de rede oferece serviço inadequado? E se ele perder pacotes com frequência?

O que acontece se os roteadores travarem de vez em quando?

Problemas ocorrem, é isso. Os usuários não têm controle real sobre a rede camada, então eles não podem resolver o problema de mau serviço usando melhores roteadores ou colocando mais tratamento de erros na camada de enlace de dados porque eles não possuem a rotas. A única possibilidade é colocar em cima da camada de rede outra camada que melhora a qualidade do serviço. Se, em uma rede sem conexão, os pacotes são perdido ou mutilado, a entidade de transporte pode detectar o problema e compensá-lo usando retransmissões. Se, em uma rede orientada a conexão, uma entidade de transporte é informado no meio de uma longa transmissão que sua conexão de rede tem

foi encerrado abruptamente, sem nenhuma indicação do que aconteceu com a corrente de dados atualmente em trânsito, ele pode configurar uma nova conexão de rede para o transporte remoto entidade. Usando esta nova conexão de rede, ele pode enviar uma consulta ao seu par, perguntando quais dados chegaram e quais não, e sabendo onde estavam, pegaram onde parou.

Em essência, a existência da camada de transporte torna possível para o trans-serviço esportivo para ser mais confiável do que a rede subjacente. Além disso, o primitivas de transporte podem ser implementadas como chamadas para procedimentos de biblioteca para fazer

eles independentes das primitivas de rede. As chamadas de serviço de rede podem variar consideravelmente de uma rede para outra (por exemplo, chamadas baseadas em uma rede sem conexão

Ethernet pode ser bastante diferente de chamadas em uma rede WiMAX orientada para conexão trabalho). Ocultar o serviço de rede por trás de um conjunto de primitivas de serviço de transporte garante que mudar a rede requer apenas a substituição de um conjunto de biblioteca procedimentos com outro que faz a mesma coisa com um subjacente diferente serviço.

Graças à camada de transporte, os programadores de aplicativos podem escrever códigos de acordo com

a um conjunto padrão de primitivas e fazer com que esses programas funcionem em uma ampla variedade

de redes, sem ter que se preocupar em lidar com diferentes inter-faces e níveis de confiabilidade. Se todas as redes reais fossem perfeitas e todas tivessem o mesmos primitivos de serviço e foram garantidos nunca, nunca alterar, o transporte camada pode não ser necessária. No entanto, no mundo real, ele cumpre a função principal de isolar as camadas superiores da tecnologia, design e imperfeições do rede.

Por esta razão, muitas pessoas fizeram uma distinção qualitativa entre leigos ers 1 a 4 de um lado e camada (s) acima de 4 do outro. O fundo quatro camadas podem ser vistas como o **provedor de serviços de transporte**, enquanto o superior camada (s) são o **usuário do serviço de transporte**. Esta distinção de provedor versus usuário tem um impacto considerável no design das camadas e coloca a camada de transporte em uma posição-chave, uma vez que forma a principal fronteira entre o provedor e o usuário do serviço confiável de transmissão de dados. É o nível que os aplicativos veem.

6.1.2 Primitivas de serviço de transporte

Para permitir que os usuários acessem o serviço de transporte, a camada de transporte deve fornecer algumas operações para programas de aplicativos, ou seja, uma interface de serviço de transporte. Cada serviço de transporte possui sua própria interface. Nesta seção, examinaremos primeiro um serviço de transporte simples (hipotético) e sua interface para ver a essência

tials. Na seção seguinte, veremos um exemplo real.

O serviço de transporte é semelhante ao serviço de rede, mas também existem alguns diferenças importantes. A principal diferença é que o serviço de rede se destina para modelar o serviço oferecido por redes reais, verrugas e tudo. Redes reais podem perder pacotes, então o serviço de rede geralmente não é confiável.

O serviço de transporte orientado para conexão, em contraste, é confiável. Claro, redes reais não são isentas de erros, mas esse é precisamente o propósito do transporte camada - para fornecer um serviço confiável no topo de uma rede não confiável.

Por exemplo, considere dois processos em uma única máquina conectada por um pipe no UNIX (ou qualquer outro recurso de comunicação entre processos). Eles assumem a conexão entre eles é 100% perfeita. Eles não querem saber sobre confirmações, pacotes perdidos, congestionamento ou qualquer coisa assim. o que que desejam é uma conexão 100% confiável. O processo *A* coloca os dados em uma extremidade do tubo e o processo *B* o retira do outro. Isso é o que o orientado a conexão serviço de transporte tem tudo a ver - esconder as imperfeições do serviço de rede para que os processos do usuário podem simplesmente assumir a existência de um fluxo de bits livre de erros, mesmo

quando eles estão em máquinas diferentes.

Como um aparte, a camada de transporte também pode fornecer serviços não confiáveis (datagramas)

vice. No entanto, há relativamente pouco a dizer sobre isso além de " são datagramas, " então vamos nos concentrar principalmente no serviço de transporte orientado para conexão neste capítulo. No entanto, existem algumas aplicações, como computação cliente-servidor streaming e streaming de multimídia, que se baseiam em um serviço de transporte sem conexão, e falaremos um pouco sobre isso mais tarde.

Uma segunda diferença entre o serviço de rede e o serviço de transporte é a quem os serviços se destinam. O serviço de rede é usado apenas pelo trans-entidades esportivas. Poucos usuários escrevem suas próprias entidades de transporte e, portanto, poucos usuários ou os programas sempre veem o serviço de rede vazio. Em contraste, muitos programas (e, portanto, programadores) consulte os primitivos de transporte. Consequentemente, o serviço de transporte deve ser conveniente e fácil de usar.

Para ter uma ideia de como pode ser um serviço de transporte, considere os cinco primitivas listadas na Figura 6-2. Esta interface de transporte é realmente básica, mas dá o sabor essencial do que uma interface de transporte orientada a conexão tem para Faz. Ele permite que programas de aplicativos estabeleçam, usem e liberem conexões, o que é suficiente para muitos aplicativos.

Para ver como esses primitivos podem ser usados, considere um aplicativo com um servidor e vários clientes remotos. Para começar, o servidor executa um LISTEN primitivo, normalmente chamando um procedimento de biblioteca que faz uma chamada de sistema que

SEC. 6,1

O SERVIÇO DE TRANSPORTE

499

Primitivo

Pacote enviado

Significado

OUÇO

(Nenhum)

Bloquear até que algum processo tente se conectar

CONECTAR

CONEXÃO REQ.

Tentar estabelecer uma conexão ativamente

ENVIAR

DADOS

Envie informação

RECEBER

(Nenhum)

Bloquear até que um pacote de dados chegue
DESCONECTAR

DESCONEXÃO REQ. Solicite a liberação da conexão

Figura 6-2. Os primitivos para um serviço de transporte simples.

bloqueia o servidor até que um cliente apareça. Quando um cliente deseja falar com o servidor, ele executa uma primitiva CONNECT . A entidade de transporte realiza esta primitiva por bloquear o chamador e enviar um pacote ao servidor. Encapsulado no pagamento a carga desse pacote é uma mensagem da camada de transporte para a entidade de transporte do servidor.

Uma nota rápida sobre a terminologia está em ordem. Por falta de um termo melhor, nós usará o termo **segmento** para mensagens enviadas da entidade de transporte para transportar tide. TCP, UDP e outros protocolos da Internet usam esse termo. Alguns protocolos mais antigos usou o nome desajeitado **TPDU** (**T**ransport **P**rotocol **D**ata **U**nit). Esse termo é não é usado muito mais agora, mas você pode vê-lo em jornais e livros mais antigos. Assim, os segmentos (trocados pela camada de transporte) estão contidos em pacotes (trocado pela camada de rede). Por sua vez, esses pacotes estão contidos em frames (trocado pela camada de enlace de dados). Quando um quadro chega, a camada de enlace de dados processa o cabeçalho do quadro e, se o endereço de destino corresponder para entrega local ery, passa o conteúdo do campo de carga útil do quadro para a entidade de rede. o entidade de rede similarmente processa o cabeçalho do pacote e então passa o conteúdo da carga útil do pacote até a entidade de transporte. Este aninhamento é ilustrado em

Fig. 6-3.

Quadro, Armação
cabeçalho
Pacote
cabeçalho
Segmento
cabeçalho
Carga útil do segmento
Carga útil do quadro
Carga útil do pacote

Figura 6-3. Aninhamento de segmentos, pacotes e quadros.

Voltando ao nosso exemplo cliente-servidor, a chamada CONNECT do cliente causa um CONNECTION REQUEST segmento a ser enviado ao servidor. Quando chega, o

Página 524

500

A CAMADA DE TRANSPORTE

INDIVÍDUO. 6

a entidade de transporte verifica se o servidor está bloqueado em um LISTEN (ou seja, está inter- estabelecido no tratamento de solicitações). Nesse caso, ele desbloqueia o servidor e envia um CONNECTION ACCEPTED segmento de volta ao cliente. Quando este segmento chega, o cliente é desbloqueado e a conexão é estabelecida.

Os dados agora podem ser trocados usando as primitivas SEND e RECEIVE . No forma mais simples, qualquer uma das partes pode fazer um (bloqueando) RECEIVE para aguardar a outra parte

para fazer um ENVIAR . Quando o segmento chega, o receptor é desbloqueado. Pode então processar o segmento e enviar uma resposta. Contanto que ambos os lados possam acompanhar de quem é a vez de enviar, esse esquema funciona bem.

Observe que na camada de transporte, mesmo uma simples troca de dados unidirecional é mais complicado do que na camada de rede. Cada pacote de dados enviado também será reconhecido (eventualmente). Os pacotes contendo segmentos de controle também são reconhecido, implícita ou explicitamente. Esses reconhecimentos são gerenciados por as entidades de transporte, usando o protocolo da camada de rede, e não são visíveis para o usuários de transporte. Da mesma forma, as entidades de transporte precisam se preocupar com temporizadores e

retransmissões. Nenhuma dessas máquinas é visível para os usuários do transporte. Ao usuários de transporte, uma conexão é um canal de bits confiável: um usuário insere bits e eles magicamente aparecem na mesma ordem na outra extremidade. Esta capacidade de ocultar plexidade é a razão pela qual os protocolos em camadas são uma ferramenta tão poderosa. Quando uma conexão não é mais necessária, ela deve ser liberada para liberar mesa

espaço dentro das duas entidades de transporte. A desconexão tem duas variantes: assymétrico e simétrico. Na variante assimétrica, qualquer usuário de transporte pode emitir um Primitiva DISCONNECT , que resulta em um segmento DISCONNECT sendo enviado para o entidade de transporte remoto. Após sua chegada, a conexão é liberada.

Na variante simétrica, cada direção é fechada separadamente, independentemente de o outro. Quando um lado faz um DESCONECTAR , isso significa que não tem mais dados enviar, mas ainda está disposto a aceitar dados de seu parceiro. Neste modelo, um conexão é liberada quando ambos os lados fizeram uma DESCONEXÃO .

Um diagrama de estado para estabelecimento e liberação de conexão para estes as primitivas são apresentadas na Figura 6-4. Cada transição é desencadeada por algum evento, seja um primitivo executado pelo usuário de transporte local ou um pacote de entrada. Para simplicidade, assumimos aqui que cada segmento é reconhecido separadamente. Nós também as suponha que um modelo de desconexão simétrica seja usado, com o cliente primeiro.

Observe que este modelo não é sofisticado. Veremos mais realis-modelos tíc mais tarde, quando descrevermos como o TCP funciona.

6.1.3 Soquetes Berkeley

Vamos agora inspecionar brevemente outro conjunto de primitivos de transporte, o soquete prim-ativos como são usados para TCP. Os soquetes foram lançados pela primeira vez como parte do Berke-distribuição de software ley UNIX 4.2BSD em 1983. Eles se tornaram populares rapidamente. As primitivas são agora amplamente utilizadas para programação de Internet em muitas

Página 525

SEC. 6.1
O SERVIÇO DE TRANSPORTE

501
ATIVO
ESTABELECIMENTO
PENDENTE
PASSIVA
ESTABELECIMENTO
PENDENTE
PASSIVA
DESCONECTAR
PENDENTE
ATIVO
DESCONECTAR
PENDENTE
OCIOSO
OCIOSO
ESTABELECIDO
Desconexão
solicitar segmento
recebido
desconectar
primitivo
executado
desconectar
primitivo executado
Pedido de desconexão
segmento recebido
Pedido de conexão
segmento recebido
Conexão aceita
segmento recebido
Conectar primitivo
executado
Conectar primitivo
executado

Figura 6-4. Um diagrama de estado para um esquema simples de gerenciamento de conexão.
As transições marcadas em itálico são causadas por chegadas de pacotes. As linhas sólidas mostram a sequência de estado do cliente. As linhas tracejadas mostram a sequência de estado do servidor. sistemas, especialmente sistemas baseados em UNIX , e há uma API estilo socket para Win-dows chamado " winsock. "

As primitivas estão listadas na Figura 6-5. Grosso modo, eles seguem o mo-del de nosso primeiro exemplo, mas oferece mais recursos e flexibilidade. Não vamos olhar nos segmentos correspondentes aqui. Essa discussão virá mais tarde.

Primitivo

Significado
SOQUETE
Crie um novo endpoint de comunicação
LIGAR
Associar um endereço local a um soquete
OUÇO
Anuncie a disposição de aceitar conexões; dar o tamanho da fila
ACEITAR
Estabeleça passivamente uma conexão de entrada
CONECTAR
Tentar estabelecer uma conexão ativamente
ENVIAR
Envie alguns dados pela conexão
RECEBER
Receba alguns dados da conexão
FECHAR
Libere a conexão

Figura 6-5. As primitivas de soquete para TCP.

Página 526

502

A CAMADA DE TRANSPORTE

INDIVÍDUO. 6

As primeiras quatro primitivas na lista são executadas nessa ordem pelos servidores. A primitiva SOCKET cria um novo ponto de extremidade e aloca espaço de tabela para ele dentro a entidade de transporte. Os parâmetros da chamada especificam o formato de endereçamento para ser usado, o tipo de serviço desejado (por exemplo, fluxo de bytes confiável) e o protocolo.

Uma chamada SOCKET bem-sucedida retorna um descritor de arquivo comum para uso em processos chamadas, da mesma forma que uma chamada OPEN em um arquivo.

Os soquetes recém-criados não têm endereços de rede. Estes são atribuídos usando a primitiva BIND . Depois que um servidor vincula um endereço a um soquete, os clientes podem se conectar a ele. O motivo de não ter a chamada SOCKET criando um anúncio vestir-se diretamente é que alguns processos se preocupam com seus endereços (por exemplo, eles têm

usa o mesmo endereço há anos e todo mundo conhece esse endereço), enquanto outros não fazem.

Em seguida, vem a chamada LISTEN , que aloca espaço para enfileirar as chamadas de entrada para o caso de vários clientes tentarem se conectar ao mesmo tempo. Em contraste com LISTEN em nosso primeiro exemplo, no modelo de socket, LISTEN não é uma chamada de bloqueio.

Para bloquear a espera por uma conexão de entrada, o servidor executa um ACCEPT primitivo. Quando chega um segmento pedindo uma conexão, a entidade de transporte cria um novo soquete com as mesmas propriedades do original e retorna um arquivo descritor para ele. O servidor pode então bifurcar um processo ou thread para lidar com a conexão no novo soquete e volte a esperar pela próxima conexão no soquete original. ACCEPT retorna um descritor de arquivo, que pode ser usado para leitura escrever e escrever na forma padrão, o mesmo que para arquivos.

Agora, vejamos o lado do cliente. Aqui, também, um soquete deve primeiro ser criado usando a primitiva SOCKET , mas BIND não é necessário, pois o endereço usado não importa para o servidor. A primitiva CONNECT bloqueia o chamador e ativamente inicia o processo de conexão. Quando for concluído (ou seja, quando o segmento apropriado é recebido do servidor), o processo do cliente é desbloqueado e a conexão é estabelecida. Ambos os lados agora podem usar SEND e RECEIVE para transmitir e receber dados pela conexão full-duplex. O padrão UNIX READ e WRITE chamadas de sistema também podem ser usadas se nenhuma das opções especiais de ENVIAR e RECEBER é requerido.

A liberação da conexão com soquetes é simétrica. Quando ambos os lados executam cortada uma primitiva CLOSE , a conexão é liberada.

Os soquetes provaram ser extremamente populares e são o padrão de fato para abstraindo serviços de transporte para aplicativos. A API de soquete é freqüentemente usada com

o protocolo TCP para fornecer um serviço orientado a conexão denominado **byte confiável stream**, que é simplesmente o bit pipe confiável que descrevemos. No entanto, outros protocolos podem ser usados para implementar este serviço usando a mesma API. Deveria todos sejam iguais para os usuários dos serviços de transporte.

Um ponto forte da API de soquete é que ela pode ser usada por um aplicativo para outros serviços de transporte. Por exemplo, os soquetes podem ser usados com um trans-serviço desportivo. Neste caso, CONNECT define o endereço do par de transporte remoto e ENVIAR e RECEBER enviam e recebem datagramas de e para o par remoto.

Página 527

SEC. 6,1

O SERVIÇO DE TRANSPORTE

503

(Também é comum usar um conjunto expandido de chamadas, por exemplo, SENDTO e RECEIVEFROM, que enfatizam as mensagens e não limitam um aplicativo a um pecado gle transport peer.) Sockets também podem ser usados com protocolos de transporte que fornecem um fluxo de mensagens em vez de um fluxo de bytes e que tem ou não congestionamento ao controle. Por exemplo, **DCCP (Datagram Congestion Controlled Protocol)** é uma versão do UDP com controle de congestionamento (Kohler et al., 2006). Depende do tran-usuários de esportes para entender o serviço que estão recebendo.

No entanto, os soquetes provavelmente não serão a palavra final em interfaces de transporte. Por exemplo, os aplicativos geralmente funcionam com um grupo de fluxos relacionados, como um Navegador da Web que solicita vários objetos do mesmo servidor. Com soquetes, o o ajuste mais natural é que os programas de aplicativos usem um fluxo por objeto. Isto estrutura significa que o controle de congestionamento é aplicado separadamente para cada fluxo, não

em todo o grupo, o que é inferior ao ideal. Isso impõe ao aplicativo o fardo de gerenciar o conjunto. Novos protocolos e interfaces foram desenvolvidos para oferecer suporte grupos de fluxos relacionados de forma mais eficaz e simples para o aplicativo. Dois exemplos são **SCTP (Stream Control Transmission Protocol)** definido em RFC 4960 e **SST (Structured Stream Transport)** (Ford, 2007). Estes protocolos deve alterar ligeiramente a API de soquete para obter os benefícios dos grupos de streams, e eles também suportam recursos como uma combinação de orientado a conexão e tráfego sem conexão e até mesmo vários caminhos de rede. O tempo dirá se eles são bem sucedido.

6.1.4 Um exemplo de programação de soquete: um servidor de arquivos da Internet

Como um exemplo de como as chamadas de soquete reais são feitas, considere o código do cliente e do servidor da Figura 6-6. Aqui temos um arquivo de Internet muito primitivo servidor junto com um cliente de exemplo que o utiliza. O código tem muitas limitações (discutido abaixo), mas, em princípio, o código do servidor pode ser compilado e executado em qualquer sistema UNIX conectado à Internet. O código do cliente pode ser compilado e executado em qualquer outra máquina UNIX na Internet, em qualquer lugar do mundo. O cli-código ent pode ser executado com parâmetros apropriados para buscar qualquer arquivo para o qual o servidor tem acesso em sua máquina. O arquivo é gravado na saída padrão, que, claro, pode ser redirecionado para um arquivo ou canal.

Vejamos primeiro o código do servidor. Ele começa incluindo algum padrão cabeçalhos, os últimos três dos quais contêm as principais definições relacionadas à Internet e estruturas de dados. Em seguida, vem uma definição de **PORTA DO SERVIDOR** como 12345. Este número

ber foi escolhido arbitrariamente. Qualquer número entre 1024 e 65535 funcionará apenas também, contanto que não esteja em uso por algum outro processo; portas abaixo de 1023 são reservado para usuários privilegiados.

As próximas duas linhas no servidor definem duas constantes necessárias. O primeiro determina o tamanho do bloco em bytes usado para a transferência do arquivo. O segundo de-encerra quantas conexões pendentes podem ser mantidas antes que as adicionais sejam descartados na chegada.

504A CAMADA DE TRANSPORTE
INDIVÍDUO. 6

```

/* Esta página contém um programa cliente que pode solicitar um arquivo do programa servidor
 * na próxima página. O servidor responde enviando o arquivo inteiro.
 */
#include <sys / types.h>
#include <sys / socket.h>
#include <netinet / in.h>
#include <netdb.h>
#define PORTA DO SERVIDOR 12345
/* arbitrário, mas cliente e servidor devem concordar */
#define TAMANHO BUF 4096
/* tamanho de transferência de bloco */
int main (int argc, char ** argv)
{
    int c, s, bytes;
    char buf [BUF SIZE];
    /* buffer para arquivo de entrada */
    struct hostent * h;
    /* informações sobre o servidor */
    struct sockaddr no canal;
    /* contém o endereço IP */
    if (argc! = 3) fatal ("Uso: cliente nome do servidor nome do arquivo");
    h = gethostbyname (argv [1]);
    /* procurar o endereço IP do host */
    if (! h) fatal ("gethostbyname falhou");
    s = socket (PF INET, SOCK STREAM, IPPROTO TCP);
    if (s <0) fatal ("socket");
    memset (& canal, 0, sizeof (canal));
    channel.sin family = AF INET;
    memcpy (& channel.sin addr.s addr, h-> h addr, h-> h comprimento);
    channel.sin porta = htons (PORTA DO SERVIDOR);
    c = conectar (s, (struct sockaddr * ) & canal, sizeof (canal));
    if (c <0) fatal ("falha na conexão");
    /* A conexão agora está estabelecida. Envie o nome do arquivo incluindo 0 byte no final. */
    escrever (s, argv [2], strlen (argv [2]) + 1);
    /* Vá obter o arquivo e grave-o na saída padrão. */
    enquanto (1) {
        bytes = leitura (s, buf, BUF SIZE);
        /* ler do soquete */
        if (bytes <= 0) exit (0);
        /* verificar o fim do arquivo */
        escrever (1, buf, bytes);
        /* escrever na saída padrão */
    }
    fatal ( string * char )
    {
        printf ("%s\n", string);
        saída (1);
    }

```

Figura 6-6. Código do cliente usando soquetes. O código do servidor está na próxima página.

SEC. 6,1

O SERVIÇO DE TRANSPORTE

505

```

#include <sys / types.h>
/* Este é o código do servidor */

```

```

#include <sys / fcntl.h>
#include <sys / socket.h>
#include <netinet / in.h>
#include <netdb.h>
#define PORTA DO SERVIDOR 12345
/* arbitrário, mas cliente e servidor devem concordar */
#define TAMANHO BUF 4096
/* tamanho de transferência de bloco */
#define QUEUE SIZE 10
int main (int argc, char * argv [])
{
int s, b, l, fd, sa, bytes, on = 1;
char buf [BUF SIZE];
/* buffer para arquivo de saída */
struct sockaddr no canal;
/* contém o endereço IP */
/* Construir estrutura de endereço para vincular ao soquete. */
memset (& canal, 0, sizeof (canal));
/* canal zero */
channel.sin family = AF INET;
channel.sin addr.s addr = htonl (INADDR ANY);
channel.sin porta = htons (PORTA DO SERVIDOR);
/* Passivo aberto. Aguarde a conexão. */
s = socket (AF INET, SOCK STREAM, IPPROTO TCP); /* criar soquete */
if (s <0) fatal ("socket falhou");
setsockopt (s, SOL SOCKET, SO REUSEADDR, (char *) & on, sizeof (on));
b = vincular (s, (struct sockaddr *) & canal, sizeof (canal));
if (b <0) fatal ("falha de ligação");
l = ouvir (s, TAMANHO DA FILA);
/* especificar o tamanho da fila */
if (l <0) fatal ("escuta falhou");
/* O soquete agora está configurado e vinculado. Aguarde a conexão e processe-a. */
enquanto (1) {
sa = aceitar (s, 0, 0);
/* bloco para solicitação de conexão */
if (sa <0) fatal ("falha na aceitação");
ler (sa, buf, BUF SIZE);
/* ler o nome do arquivo do soquete */
/* Pega e retorna o arquivo. */
fd = aberto (buf, O RDONLY);
/* abre o arquivo a ser enviado de volta */
if (fd <0) fatal ("falha ao abrir");
enquanto (1) {
bytes = leitura (fd, buf, BUF SIZE); /* ler do arquivo */
se (bytes <= 0) quebrar;
/* verificar o fim do arquivo */
escrever (sa, buf, bytes);
/* escrever bytes no soquete */
}
fechar (fd);
/* fechar arquivo */
fechar (sa);
/* fechar conexão */
}
}

```

inicializando uma estrutura de dados que conterá o endereço IP do servidor. Estes dados a estrutura em breve será ligada ao soquete do servidor. A chamada para *memset* define o estrutura de dados para todos os 0s. As três atribuições seguintes preenchem três de seus Campos. O último deles contém a porta do servidor. As funções *htonl* e *htons* tem a ver com a conversão de valores para um formato padrão para que o código seja executado corretamente em máquinas little-endian (por exemplo, Intel x86) e máquinas big-endian (por exemplo, o SPARC). Sua semântica exata não é relevante aqui.

Em seguida, o servidor cria um soquete e verifica os erros (indicados por $s < 0$). No uma versão de produção do código, a mensagem de erro poderia ser um pouco mais explicativa história. A chamada para *setsockopt* é necessária para permitir que a porta seja reutilizada para que o servidor pode ser executado indefinidamente, atendendo solicitação após solicitação. Agora o endereço IP está vinculado a o soquete e uma verificação é feita para ver se a chamada para *vincular* foi bem-sucedida. A etapa final na inicialização é a chamada para *ouvir* para anunciar a vontade do servidor de aceitar receba chamadas e diga ao sistema para reter até *TAMANHO DE FILA* delas em caso novos pedidos cheguem enquanto o servidor ainda está processando o atual. Se o fila está cheia e chegam pedidos adicionais, eles são descartados silenciosamente.

Nesse ponto, o servidor entra em seu loop principal, do qual nunca sai. O único maneira de pará-lo é matá-lo de fora. A chamada para *aceitar* bloqueia o servidor até algum cliente tenta estabelecer uma conexão com ele. Se a chamada de *aceitação* for bem-sucedida, retorna um descritor de socket que pode ser usado para ler e escrever, análogo a como os descritores de arquivo podem ser usados para ler e gravar nos canais. No entanto, ao contrário tubos, que são unidirecionais, os soquetes são bidirecionais, então *sa* (o socket) pode ser usado para ler a partir da conexão e também para escrever nela. UMA O descritor de arquivo pipe serve para leitura ou escrita, mas não para ambos.

Depois que a conexão é estabelecida, o servidor lê o nome do arquivo dele. E se o nome ainda não está disponível, o servidor bloqueia esperando por ele. Depois de obter o nome do arquivo, o servidor abre o arquivo e entra em um loop que lê blocos alternadamente do arquivo e os grava no soquete até que todo o arquivo seja copiado.

Em seguida, o servidor fecha o arquivo e a conexão e aguarda a próxima confirmação para aparecer. Ele repete esse loop para sempre.

Agora, vamos examinar o código do cliente. Para entender como funciona, é necessário É necessário entender como ele é invocado. Supondo que seja chamado de *cliente*, uma chamada típica é

```
cliente flits.cs.vu.nl / usr / tom / nome do arquivo> f
```

Esta chamada só funciona se o servidor já estiver rodando em *flits.cs.vu.nl* e o arquivo */usr / tom / filename* existe e o servidor tem acesso de leitura a ele. Se a chamada for bem-sucedida sucesso, o arquivo é transferido pela Internet e gravado em *f*, apesar o qual o cli- saídas do programa ent. Uma vez que o servidor continua após uma transferência, o cliente pode ser começou novamente e novamente para obter outros arquivos.

O código do cliente começa com algumas inclusões e declarações. Execução começa verificando se ele foi chamado com o número certo de argumentos (*argc* = 3 significa o nome do programa mais dois argumentos). Observe que *argv [1]* contém o

estabelecer uma conexão TCP com o servidor, usando *conectar*. Se o servidor estiver ligado e rodando

ning na máquina nomeada e conectado à *PORTA DO SERVIDOR* e está ocioso ou tem espaço em sua fila de *escuta*, a conexão será (eventualmente) estabelecida.

Usando a conexão, o cliente envia o nome do arquivo escrevendo no socket. O número de bytes enviados é um maior que o nome propriamente dito, uma vez que o 0 byte que encerra o nome também deve ser enviado para informar ao servidor onde o nome termina.

Agora o cliente entra em um loop, lendo o arquivo bloco por bloco do socket e copiá-lo para a saída padrão. Quando estiver pronto, ele simplesmente sai.

O procedimento *fatal* imprime uma mensagem de erro e sai. O servidor precisa do mesmo procedimento, mas foi omitido por falta de espaço na página. Desde o cliente e servidor são compilados separadamente e normalmente executados em diferentes computadores

computadores, eles não podem compartilhar o código de *fatal*.

Esses dois programas (bem como outros materiais relacionados a este livro) podem ser obtido no site do livro

<http://www.pearsonhighered.com/tanenbaum>

Apenas para registro, este servidor não é a última palavra em serverdom. Seu erro a verificação é insuficiente e seu relatório de erros é medíocre. Uma vez que lida com todos os re-missões estritamente sequenciais (porque tem apenas um único thread), seu desempenho é pobre. Ele claramente nunca ouviu falar sobre segurança e usando chamadas de sistema UNIX simples

não é a maneira de obter independência de plataforma. Também faz algumas suposições que são tecnicamente ilegais, como assumir que o nome do arquivo cabe no buffer e é transmitido atomicamente. Apesar dessas deficiências, é um trabalho servidor de arquivos da Internet. Nos exercícios, o leitor é convidado a aprimorá-lo. Para mais informações sobre programação com soquetes, consulte Donahoo e Calvert (2008, 2009).

6.2 ELEMENTOS DE PROTOCOLOS DE TRANSPORTE

O serviço de transporte é implementado por um **protocolo de transporte** usado entre as duas entidades de transporte. De certa forma, os protocolos de transporte se parecem com os dados

protocolos de link que estudamos em detalhes no Cap. 3. Ambos têm que lidar com erros de controle, sequenciamento e controle de fluxo, entre outras questões.

No entanto, também existem diferenças significativas entre os dois. Estes diferenças são devido a grandes diferenças entre os ambientes em que o dois protocolos operam, conforme mostrado na Figura 6-7. Na camada de enlace de dados, dois roteadores

Página 532

508

A CAMADA DE TRANSPORTE
INDIVÍDUO. 6

comunicar-se diretamente por meio de um canal físico, com ou sem fio, enquanto na camada de transporte, esse canal físico é substituído por toda a rede. Isto diferença tem muitas implicações importantes para os protocolos.

Roteador
Roteador
Física
canal de comunicação
Hospedeiro
(uma)
(b)
Rede

Figura 6-7. (a) Ambiente da camada de enlace. (b) Meio Ambiente da camada de transporte.

Por um lado, em links ponto a ponto, como fios ou fibra óptica, é geralmente não é necessário que um roteador especifique com qual roteador deseja se comunicar - cada

a linha de saída leva diretamente a um determinado roteador. Na camada de transporte, explícito o endereçamento de destinos é necessário.

Por outro lado, o processo de estabelecer uma conexão através do fio de A Fig. 6-7 (a) é simples: a outra extremidade está sempre lá (a menos que tenha travado, na qual caso não esteja lá). De qualquer forma, não há muito o que fazer. Mesmo em links sem fio, o processo não é muito diferente. Basta enviar uma mensagem para tê-la alcance todos os outros destinos. Se a mensagem não for confirmada devido a um erro, pode ser reenviado. Na camada de transporte, o estabelecimento da conexão inicial é complicado ed, como veremos.

Outra diferença (extremamente irritante) entre a camada de enlace de dados e a camada de transporte é a existência potencial de capacidade de armazenamento na rede.

Quando um roteador envia um pacote por um link, ele pode chegar ou ser perdido, mas não pode pular por um tempo, se esconder em um canto distante do mundo, e de repente surgem depois de outros pacotes que foram enviados muito mais tarde. Se a rede usa dadas-gramas, que são roteados de forma independente para dentro, há uma probabilidade não desprezível que um pacote pode tomar a rota cênica e chegar atrasado e fora do esperado pedido, ou mesmo que cheguem duplicados do pacote. As consequências do a capacidade da rede de atrasar e duplicar pacotes às vezes pode ser desastrosa e pode exigir o uso de protocolos especiais para transportar informações corretamente.

Uma diferença final entre o link de dados e as camadas de transporte é de grau ao invés de espécie. Buffer e controle de fluxo são necessários em ambas as camadas, mas o presença na camada de transporte de um grande e variado número de conexões com largura de banda que flutua conforme as conexões competem entre si pode exigir uma abordagem diferente da que usamos na camada de enlace de dados. Alguns dos protocolos discutido no cap. 3 alocar um número fixo de buffers para cada linha, de modo que quando um quadro chega, um buffer está sempre disponível. Na camada de transporte, o maior número número de conexões que devem ser gerenciadas e variações na largura de banda de cada

Página 533

SEC. 6,2

ELEMENTOS DE PROTOCOLOS DE TRANSPORTE

509

a conexão pode receber crie a ideia de dedicar muitos buffers a cada um a menos atraente. Nas seções a seguir, examinaremos todos esses importantes processa e outros.

6.2.1 Endereçamento

Quando um processo de aplicativo (por exemplo, um usuário) deseja configurar uma conexão com um

processo de aplicativo remoto, ele deve especificar a qual conectar. (Vigarista-transporte sem conexão tem o mesmo problema: para quem cada mensagem deve ser enviado?) O método normalmente usado é definir endereços de transporte para os quais processos podem ouvir solicitações de conexão. Na Internet, esses terminais são chamados portas . Usaremos o termo genérico **TSAP (Ponto de Acesso do Serviço de Transporte)** para significa um ponto final específico na camada de transporte. Os pontos de extremidade análogos na camada de rede (ou seja, endereços de camada de rede) não são surpreendentemente chamados de **NSAPs**

(**Pontos de acesso do serviço de rede**). Os endereços IP são exemplos de NSAPs.

A Figura 6-8 ilustra a relação entre os NSAPs, os TSAPs e um conexão de transporte. Os processos de aplicativos, tanto clientes quanto servidores, podem conectar para um TSAP local para estabelecer uma conexão com um TSAP remoto. Estes as conexões são executadas por meio de NSAPs em cada host, conforme mostrado. O propósito de ter

TSAPs é que em algumas redes, cada computador tem um único NSAP, então de alguma forma é necessário para distinguir vários terminais de transporte que compartilham esse NSAP.

Inscrição
processo
Inscrição
camada
Transporte

```
conexão
TSAP 1522
TSAP 1208
NSAP
NSAP
Transporte
camada
Rede
camada
Link de dados
camada
Física
camada
Servidor1
Host 1
Host 2
Servidor 2
TSAP1836
```

Figura 6-8. TSAPs, NSAPs e conexões de transporte.

Página 534

510

A CAMADA DE TRANSPORTE

INDIVÍDUO. 6

Um cenário possível para uma conexão de transporte é o seguinte:

1. Um processo do servidor de e-mail se anexa ao TSAP 1522 no host 2 para esperar para uma chamada recebida. Como um processo se liga a um TSAP está fora de questão lado do modelo de rede e depende inteiramente da operação local sistema operacional. Uma chamada como nossa LISTEN pode ser usada, por exemplo.
2. Um processo de inscrição no host 1 deseja enviar uma mensagem de e-mail, então ele se anexa ao TSAP 1208 e emite uma solicitação CONNECT . o pedido especifica TSAP 1208 no host 1 como a fonte e TSAP 1522 no host 2 como o destino. Esta ação, em última análise, resulta em uma transconexão esportiva sendo estabelecida entre o processo de inscrição e o servidor.
3. O processo de inscrição é enviado pela mensagem de correio.
4. O servidor de e-mail responde informando que entregará a mensagem.
5. A conexão de transporte é liberada.

Observe que pode haver outros servidores no host 2 que estão conectados a outros TSAPs e estão aguardando conexões de entrada que chegam pelo mesmo NSAP.

A imagem pintada acima é boa, exceto que resolvemos um pequeno problema debaixo do tapete: como o processo do usuário no host 1 sabe que o servidor de e-mail está em anexoado ao TSAP 1522? Uma possibilidade é que o servidor de e-mail tenha anexado para o TSAP 1522 por anos e gradualmente todos os usuários da rede aprenderam esta. Neste modelo, os serviços têm endereços TSAP estáveis listados em arquivos em lugares bem conhecidos. Por exemplo, o arquivo */etc/services* nas listas de sistemas UNIX quais servidores estão permanentemente conectados a quais portas, incluindo o fato de que o servidor de correio está localizado na porta TCP 25.

Embora os endereços TSAP estáveis funcionem para um pequeno número de serviços principais que nunca muda (por exemplo, o servidor Web), os processos do usuário, em geral, muitas vezes querem conversar

para outros processos de usuário que não têm endereços TSAP que são conhecidos em advance, ou que possa existir por pouco tempo.

Para lidar com essa situação, um esquema alternativo pode ser usado. Neste esquema, existe um processo especial chamado **portmapper** . Para encontrar o endereço TSAP correspondente a um determinado nome de serviço, como " BitTorrent ", um usuário configura uma configuração

conexão com o portmapper (que escuta um TSAP bem conhecido). O usuário então envia uma mensagem especificando o nome do serviço, e o portmapper envia de volta o Endereço TSAP. Então o usuário libera a conexão com o portmapper e apresenta um novo com o serviço pretendido.

Neste modelo, quando um novo serviço é criado, ele deve se registrar com o portmapper, fornecendo seu nome de serviço (normalmente, uma string ASCII) e seu TSAP. O portmapper registra essas informações em seu banco de dados interno para que

quando as perguntas vierem mais tarde, ele saberá as respostas.

Página 535

SEC. 6,2

ELEMENTOS DE PROTOCOLOS DE TRANSPORTE

511

A função do portmapper é análoga à de um diretório de assistência operador no sistema telefônico - fornece um mapeamento de nomes em números. Assim como no sistema telefônico, é imprescindível que o endereço do conhecido O TSAP usado pelo portmapper é bem conhecido. Se você não conhece o número do operador de informações, você não pode chamar o operador de informações para Descubra. Se você acha que o número que você disca para obter informações é óbvio, tente em um país estrangeiro algum dia.

Muitos dos processos do servidor que podem existir em uma máquina serão usados apenas raramente. É um desperdício ter cada um deles ativo e ouvindo um TSAP estável endereço o dia todo. Um esquema alternativo é mostrado na Fig. 6-9 de uma forma simplificada Formato. É conhecido como **protocolo de conexão inicial**. Em vez de todo conceito servidor capaz de ouvir em um TSAP bem conhecido, cada máquina que deseja oferecer serviços para usuários remotos tem um **servidor de processo** especial que atua como proxy para menos

servidores muito usados. Este servidor é denominado *inetd* em sistemas UNIX . Ouve um conjunto de portas ao mesmo tempo, aguardando uma solicitação de conexão. Usuários potenciais de um serviço começa fazendo uma solicitação CONNECT , especificando o endereço TSAP do serviço que eles querem. Se nenhum servidor estiver esperando por eles, eles obtêm uma conexão com o servidor de processo, conforme mostrado na Figura 6-9 (a).

Camada
4
TSAP
Enviar
servidor
(uma)
(b)
Host 1
Host 2
Host 1
Host 2
Processo
servidor
Do utilizador
Processo
servidor
Do utilizador

Figura 6-9. Como um processo de usuário no host 1 estabelece uma conexão com um e-mail servidor no host 2 por meio de um servidor de processo.

Depois de obter o pedido de entrada, o servidor de processo gera o pedido servidor, permitindo que ele herde a conexão existente com o usuário. O novo servidor

Página 536

512

A CAMADA DE TRANSPORTE

INDIVÍDUO. 6

faz o trabalho solicitado, enquanto o servidor de processo volta a ouvir novos solicitações, conforme mostrado na Figura 6-9 (b). Este método só é aplicável quando servidores pode ser criado sob demanda.

6.2.2 Estabelecimento de Conexão

Estabelecer uma conexão parece fácil, mas na verdade é surpreendentemente complicado. À primeira vista, pareceria suficiente para uma entidade de transporte apenas enviar um CONNECTION REQUEST segmento para o destino e aguarde por uma CONNECTION Resposta ACEITADA . O problema ocorre quando a rede pode perder, atrasar, corromper, e pacotes duplicados. Esse comportamento causa complicações graves.

Imagine uma rede tão congestionada que dificilmente os reconhecimentos volte no tempo e cada pacote atinge o tempo limite e é retransmitido duas ou três vezes. Suponha que a rede usa datagramas dentro e que cada pacote segue um rota diferente. Alguns dos pacotes podem ficar presos em um engarrafamento dentro do rede e demoram muito para chegar. Ou seja, eles podem estar atrasados na rede funcionam e aparecem muito mais tarde, quando o remetente pensa que eles se perderam. O pior pesadelo possível é o seguinte. Um usuário estabelece uma conexão com um banco, envia mensagens dizendo ao banco para transferir uma grande quantidade de dinheiro para a conta de uma pessoa não totalmente confiável. Infelizmente, os pacotes decide pegar a rota panorâmica até o destino e sair explorando um remoto canto da rede. O remetente atinge o tempo limite e envia todos novamente. Isto vez que os pacotes tomam a rota mais curta e são entregues rapidamente para que o remetente volte aluga a conexão. Infelizmente, eventualmente, o lote inicial de pacotes finalmente saiu do modo escondido e chegar ao destino em ordem, pedindo ao banco para estabelecer uma nova conexão e transferência de dinheiro (novamente). O banco não tem como dizer que são duplicatas. Deve assumir que esta é uma segunda transação independente, e transfere o dinheiro novamente. Este cenário pode parecer improvável, ou mesmo implausível, mas o ponto é este: os protocolos devem ser projetados para serem corretos em todos os casos. Apenas os casos comuns precisam ser implementado de forma eficiente para obter um bom desempenho da rede, mas o protocolo deve ser capaz de lidar com os casos incomuns sem quebrar. Se não puder, nós construíram uma rede de bom tempo que pode falhar sem aviso quando as condições ções ficam difíceis. Para o restante desta seção, estudaremos o problema da duplicação atrasada cates, com ênfase em algoritmos para estabelecer conexões de forma confiável, de modo que pesadelos como o acima não podem acontecer. O ponto crucial do problema é que as duplicatas atrasadas são consideradas pacotes novos. Não podemos prevenir pacotes sejam duplicados e atrasados. Mas se e quando isso acontecer, os pacotes devem ser rejeitados como duplicatas e não processados como pacotes novos. O problema pode ser atacado de várias maneiras, nenhuma delas muito satisfatória. Uma maneira é usar endereços de transporte descartáveis. Nesta abordagem, cada vez que um

Página 537

SEC. 6,2

ELEMENTOS DE PROTOCOLOS DE TRANSPORTE

513

endereço de transporte é necessário, um novo é gerado. Quando uma conexão é re-alugado, o endereço é descartado e nunca mais usado. Pacotes duplicados atrasados então, nunca encontrará o caminho para um processo de transporte e não poderá causar danos. Contudo, essa abordagem torna mais difícil conectar-se a um processo em primeiro lugar. Outra possibilidade é dar a cada conexão um identificador único (ou seja, um se- número de sequência incrementado para cada conexão estabelecida) escolhido pelo início tiating party e colocar em cada segmento, inclusive aquele que está solicitando a conexão. Depois que cada conexão é liberada, cada entidade de transporte pode atualizar uma listagem de tabela conexões obsoletas como pares (entidade de transporte ponto a ponto, identificador de conexão). Quando- sempre que uma solicitação de conexão chega, ela pode ser verificada na tabela para ver se pertence a uma conexão liberada anteriormente. Infelizmente, este esquema tem uma falha básica: exige que cada entidade de transporte manter uma certa quantidade de informações históricas indefinidamente. Esta história deve persistem nas máquinas de origem e de destino. Caso contrário, se uma máquina trava e perde sua memória, não saberá mais quais identificadores de conexão

já foram usados por seus pares.

Em vez disso, precisamos adotar uma abordagem diferente para simplificar o problema. Ao invés de permitindo que os pacotes vivam para sempre na rede, criamos um mecanismo para matar pacotes antigos que ainda estão mancando. Com esta restrição, o problema se torna um pouco mais gerenciável.

O tempo de vida do pacote pode ser restrito a um máximo conhecido usando um (ou mais) de as seguintes técnicas:

1. Projeto de rede restrito.
2. Colocando um contador de saltos em cada pacote.
3. Registro de data e hora de cada pacote.

A primeira técnica inclui qualquer método que evite que os pacotes entrem em loop, com combinado com alguma forma de limitação de atraso, incluindo congestionamento no (agora conhecido) caminho mais longo possível. É difícil, dado que as internets podem variar de um de cidade única a internacional em escopo. O segundo método consiste em ter o contagem de saltos inicializada com algum valor apropriado e diminuída cada vez que o pacote é encaminhado. O protocolo de rede simplesmente descarta qualquer pacote cujo salto contador torna-se zero. O terceiro método requer que cada pacote suporte o tempo que foi criado, com os roteadores concordando em descartar qualquer pacote mais antigo do que alguns hora combinada. Este último método requer que os relógios do roteador sejam sincronizados ized, o que em si é uma tarefa não trivial, e na prática um contador de saltos é um aproximação suficiente com a idade.

Na prática, precisaremos garantir não só que um pacote está morto, mas também que todos os agradecimentos a ele também estão mortos, então agora vamos introduzir um período T , que é um pequeno múltiplo do verdadeiro tempo de vida máximo do pacote. O maxi-O tempo de vida do pacote principal é uma constante conservadora para uma rede; para a Internet, é um tanto arbitrariamente considerado como sendo 120 segundos. O múltiplo é dependente do protocolo

Página 538

514

A CAMADA DE TRANSPORTE INDIVÍDUO. 6

e simplesmente tem o efeito de tornar T mais longo. Se esperarmos um tempo, T segundos após um pacote foi enviado, podemos ter certeza de que todos os vestígios dele desapareceram e que nem ele nem seus reconhecimentos aparecerão repentinamente do nada para cumprir cate questões.

Com o tempo de vida dos pacotes limitado, é possível conceber uma prática e tol forma de prova para rejeitar segmentos duplicados atrasados. O método descrito abaixo é devido a Tomlinson (1975), como refinado por Sunshine e Dalal (1978). Variantes disso são amplamente utilizados na prática, inclusive no TCP.

O coração do método é a origem rotular os segmentos com a sequência números que não serão reutilizados dentro de T segundos. O período, T , e a taxa de embalagem etos por segundo determinam o tamanho dos números de sequência. Desta forma, apenas um o pacote com um determinado número de sequência pode estar pendente a qualquer momento. Dup-lisências deste pacote ainda podem ocorrer, e devem ser descartadas pelo destino ção. No entanto, não é mais o caso de uma duplicata atrasada de um pacote antigo pode vencer um novo pacote com o mesmo número de sequência e ser aceito pelo destino em seu lugar.

Para contornar o problema de uma máquina perder toda a memória de onde estava após um acidente, uma possibilidade é exigir que as entidades de transporte fiquem inativas por T segundos

após uma recuperação. O período de inatividade deixará todos os segmentos antigos morrerem, para que o remetente possa

comece novamente com qualquer número de sequência. No entanto, em uma internetwork complexa, T

pode ser grande, então essa estratégia não é atrativa.

Em vez disso, Tomlinson propôs equipar cada host com um relógio de hora do dia.

Os relógios em hosts diferentes não precisam ser sincronizados. Cada relógio é assumido como assumir a forma de um contador binário que se incrementa em intervalos uniformes. Além disso, o número de bits no contador deve ser igual ou superior ao número de bits nos números de sequência. Por último, e mais importante, o relógio é considerado continue executando mesmo se o host cair.

Quando uma conexão é configurada, os k bits de ordem inferior do relógio são usados como número de sequência inicial de k bits. Assim, ao contrário de nossos protocolos do Cap. 3, cada connection começa a numerar seus segmentos com um número de sequência inicial diferente.

O espaço da sequência deve ser tão grande que, pelos números da sequência de tempo, envolva ao redor, segmentos antigos com o mesmo número de sequência já se foram. Esta linear a relação entre o tempo e os números da sequência inicial é mostrada na Fig. 6-10 (a). A região proibida mostra os horários em que os números de sequência do segmento são ilegais levando ao seu uso. Se algum segmento for enviado com um número de sequência neste region, poderia ser atrasado e personificar um pacote diferente com o mesmo número de sequência que será emitido um pouco mais tarde. Por exemplo, se o host travar e reinicia no tempo de 70 segundos, ele usará os números de sequência iniciais com base no relógio para retomar depois que parou; o host não começa com uma sequência inferior número na região proibida.

Uma vez que ambas as entidades de transporte concordaram com o número de sequência inicial, qualquer protocolo de janela deslizante pode ser usado para controle de fluxo de dados. Este protocolo de janela irá encontrar e descartar corretamente os pacotes duplicados após eles já terem sido

Página 539

SEC. 6,2

ELEMENTOS DE PROTOCOLOS DE TRANSPORTE

515

120
80
70
60
0
30
60
90
Tempo
(uma)
Tempo
(b)
120 150 180
0
Seqüência
números
Seqüência
números
Reiniciar depois
bater com 70
T
T
Seqüência real
números usados
2 $k - 1$
Proibido
região

Figura 6-10. (a) Os segmentos não podem entrar na região proibida. (b) O resync problema de cronização.

aceitaram. Na realidade, a curva do número de sequência inicial (mostrada pela linha grossa) não é linear, mas uma escada, já que o relógio avança em passos discretos. Para simplicidade, vamos ignorar esse detalhe.

Para manter os números de sequência de pacotes fora da região proibida, precisamos tomar cuidado em dois aspectos. Podemos ter problemas de duas maneiras distintas. Se um hospedeiro

envia muitos dados muito rápido em uma conexão recém-aberta, a sequência real a curva de número versus tempo pode aumentar mais abruptamente do que o número de sequência inicial

curva versus tempo, fazendo com que o número da sequência entre na região proibida. Para evitar que isso aconteça, a taxa máxima de dados em qualquer conexão é um segmento por tique do relógio. Isso também significa que a entidade de transporte deve esperar até o relógio passa antes de abrir uma nova conexão após um reinício de falha, para que o mesmo número seja usado duas vezes. Ambos os pontos argumentam a favor de um tique-taque curto (1 μ sec ou menos). Mas o relógio não pode marcar muito rápido em relação ao número da sequência. Para uma taxa de clock de C e um espaço de número de sequência de tamanho S , devemos ter $S/C > T$ para que os números de sequência não sejam agrupados muito rapidamente.

Entrar na região proibida por baixo, enviando muito rápido, não é o única maneira de entrar em apuros. Da Fig. 6-10 (b), vemos que a qualquer taxa de dados menos do que a taxa de clock, a curva dos números de sequência reais usados em relação ao tempo eventualmente correr para a região proibida da esquerda como os números de sequência envolverem em torno. Quanto maior for a inclinação dos números de sequência reais, mais este evento será atrasado. Evitar esta situação limita a lentidão da sequência os números podem avançar em uma conexão (ou quanto tempo as conexões podem durar).

O método baseado em relógio resolve o problema de não ser capaz de distinguir atrasou segmentos duplicados de novos segmentos. No entanto, há uma prática empecilho para usá-lo para estabelecer conexões. Uma vez que normalmente não nos lembramos números de sequência através de conexões no destino, ainda não temos como

Página 540

516

A CAMADA DE TRANSPORTE INDIVÍDUO. 6

saber se um segmento CONNECTION REQUEST contendo uma sequência inicial número é uma duplicata de uma conexão recente. Este obstáculo não existe durante um conexão porque o protocolo da janela deslizante se lembra da configuração atual número de sequência.

Para resolver este problema específico, Tomlinson (1975) introduziu o modelo **triplo aperto de mão**. Este protocolo de estabelecimento envolve uma verificação de pares com o outro que o pedido de conexão é realmente atual. O procedimento de configuração normal quando o host 1 inicia é mostrado na Fig. 6-11 (a). O Host 1 escolhe um número de sequência, x , e envia um segmento CONNECTION REQUEST que o contém para o host 2. Host 2 responde com um segmento ACK reconhecendo x e anunciando seu próprio se inicial número de sequência, y . Finalmente, o host 1 reconhece a escolha do host 2 de um se- inicial número de sequência no primeiro segmento de dados que ele envia.

Agora vamos ver como o handshake de três vias funciona na presença de atraso de segmentos de controle duplicados. Na Fig. 6-11 (b), o primeiro segmento é uma duplicação atrasada de um CONNECTION REQUEST de uma conexão antiga. Este segmento chega a host 2 sem o conhecimento do host 1. O host 2 reage a este segmento enviando host 1 um segmento ACK, solicitando a verificação de que o host 1 estava realmente tentando para configurar uma nova conexão. Quando o host 1 rejeita a tentativa do host 2 de estabelecer um conexão, o host 2 percebe que foi enganado por uma duplicação atrasada e abandona a conexão. Dessa forma, uma duplicata atrasada não causa danos.

O pior caso é quando um CONNECTION REQUEST e um ACK atrasados estão flutuando na sub-rede. Esse caso é mostrado na Fig. 6-11 (c). Como no exemplo anterior, o host 2 recebe um PEDIDO DE CONEXÃO atrasado e responde a isto. Neste ponto, é crucial perceber que o host 2 propôs usar y como o número de sequência inicial do host 2 para o tráfego do host 1, sabendo muito bem que nenhum

segmento contendo o número de sequência y ou reconhecimentos a y ainda existem. Quando o segundo segmento atrasado chega ao host 2, o fato de z ter sido reconhecido em vez de y diz ao host 2 que essa também é uma duplicata antiga. O importante perceber aqui é que não há combinação de segmentos antigos que pode fazer com que o protocolo falhe e ter uma conexão configurada acidentalmente quando não quer.

O TCP usa esse handshake de três vias para estabelecer conexões. Dentro de um con-

nection, um timestamp é usado para estender o número de sequência de 32 bits para que não empacotar dentro da vida útil máxima do pacote, mesmo para gigabit por segundo com conexões. Este mecanismo é uma correção para o TCP que era necessária, pois era usado mais rapidamente

e links mais rápidos. É descrito no RFC 1323 e denominado **PAWS (Proteção Contra números de sequência embrulhados)**. Através de conexões, para o se- inicial números de referência e antes que o PAWS pudesse entrar em jogo, o TCP originalmente usou o esquema baseado em relógio que acabamos de descrever. No entanto, isso acabou por ter uma segurança

vulnerabilidade. O relógio tornou fácil para um invasor prever a próxima sequência inicial enviar o número e enviar pacotes que enganaram o handshake de três vias e o estabelecimento lished uma conexão forjada. Para fechar este buraco, a sequência inicial pseudo-aleatória números são usados para conexões na prática. No entanto, continua sendo importante que

Página 541

SEC. 6,2

ELEMENTOS DE PROTOCOLOS DE TRANSPORTE

517

Tempo
Tempo
Tempo
DADOS (seq = x, ACK = y)
ACK (seq = y, ACK = x)
CR (seq = x)
Host 1
Host 2
REJEITAR (ACK = y)
DADOS (seq = x,
ACK = z)
ACK (seq = y, ACK = x)
CR (seq = x)
Host 1
Host 2
REJEITAR (ACK = y)
ACK (seq = y, ACK = x)
CR (seq = x)
Host 1
Host 2
Duplicata antiga
Duplicata antiga
Duplicata antiga
(uma)
(b)
(c)

Figura 6-11. Três cenários de protocolo para estabelecer uma conexão usando um aperto de mão de três vias. CR denota PEDIDO DE CONEXÃO . (a) Operação normal ção. (b) SOLICITAÇÃO DE CONEXÃO duplicada antiga aparecendo do nada.

(c) Solicitação de conexão duplicada e ACK duplicado .

os números da sequência inicial não se repetem por um intervalo, mesmo que apareçam aleatório para um observador. Caso contrário, duplicatas atrasadas podem causar estragos.

6.2.3 Liberação de Conexão

Liberar uma conexão é mais fácil do que estabelecer uma. No entanto, existem mais armadilhas do que se poderia esperar aqui. Como mencionamos anteriormente, existem dois estilos de encerrar uma conexão: liberação assimétrica e liberação simétrica.

Página 542

518

A CAMADA DE TRANSPORTE

INDIVÍDUO. 6

A liberação assimétrica é a forma como o sistema telefônico funciona: quando uma das partes trava acima, a conexão é interrompida. A versão simétrica trata a conexão como dois set arate conexões unidirecionais e requer que cada uma seja liberada separadamente.

A liberação assimétrica é abrupta e pode resultar em perda de dados. Considere o cenário ario da Fig. 6-12. Depois que a conexão é estabelecida, o host 1 envia um segmento que chega corretamente ao host 2. Em seguida, o host 1 envia outro segmento. Infelizmente, o host 2 emite um DISCONNECT antes que o segundo segmento chegue. O resultado é que

a conexão é liberada e os dados são perdidos.

Tempo
CR
DADOS
DADOS
Host 1
Host 2
ACK
DR
Sem dados são
entregue depois
uma desconexão
solicitação

Figura 6-12. Desconexão abrupta com perda de dados.

Claramente, um protocolo de lançamento mais sofisticado é necessário para evitar a perda de dados. Uma maneira é usar a liberação simétrica, em que cada direção é liberada independentemente diently do outro. Aqui, um host pode continuar a receber dados mesmo depois de ter enviado um segmento DISCONNECT .

A liberação simétrica faz o trabalho quando cada processo tem uma quantidade fixa de dados para enviar e sabe claramente quando o enviou. Em outras situações, determinar que todo o trabalho foi feito e a conexão deve ser encerrada não é tão obvious. Pode-se imaginar um protocolo no qual o host 1 diga " Pronto. Você está feito também? " Se o host 2 responder: " Eu também terminei. Adeus a conexão pode ser liberado com segurança. "

Infelizmente, esse protocolo nem sempre funciona. Há um famoso problema que ilustra esse problema. É o chamado problema dos **dois exércitos** . Imagine que um O exército branco está acampado em um vale, como mostra a Figura 6-13. Em ambos os sur-encostas arredondadas são exércitos azuis. O exército branco é maior do que qualquer um dos exércitos azuis sozinhos, mas juntos os exércitos azuis são maiores que o exército branco. E se ou o exército azul ataca por si mesmo, será derrotado, mas se os dois exércitos azuis atacarem seguir simultaneamente, eles serão vitoriosos.

Os exércitos azuis querem sincronizar seus ataques. No entanto, seu único compromisso meio de comunicação é enviar mensageiros a pé para o vale, onde

Página 543

SEC. 6,2

ELEMENTOS DE PROTOCOLOS DE TRANSPORTE

519

W
B
B
Exército branco
Azul
exército
1
Azul
exército
2

Figura 6-13. O problema dos dois exércitos.

eles podem ser capturados e a mensagem perdida (ou seja, eles têm que usar um não confiável canal de comunicação). A questão é: existe um protocolo que permite o exércitos azuis para vencer?

Suponha que o comandante do exército azul # 1 envie uma mensagem dizendo: " Eu propomos que atacemos na madrugada de 29 de março. Que tal? " Agora suponha que o mensagem chega, o comandante do exército azul # 2 concorda, e sua resposta chega com segurança de volta ao exército azul # 1. O ataque acontecerá? Provavelmente não, porque o comandante # 2 não sabe se sua resposta foi recebida. Se não, o exército azul # 1 não atacará tática, então seria tolice ele atacar a batalha.

Agora, vamos melhorar o protocolo tornando-o um handshake de três vias. o iniciador da proposta original deve reconhecer a resposta. Supondo que não mensagens são perdidas, o exército azul # 2 receberá o reconhecimento, mas o com o comandante do exército azul # 1 agora hesitará. Afinal, ele não sabe se seu ac- conhecimento passou, e se não, ele sabe que o exército azul # 2 não vai ataque. Agora poderíamos fazer um protocolo de handshake de quatro vias, mas isso não ajudar também.

Na verdade, pode-se provar que não existe nenhum protocolo que funcione. Suponha que algum protocolo existia. Ou a última mensagem do protocolo é essencial ou não é. Se não estiver, podemos removê-lo (e quaisquer outras mensagens não essenciais) até que tenhamos com um protocolo em que cada mensagem é essencial. O que acontece se a final a mensagem não chega? Acabamos de dizer que era essencial, por isso, se for perdido, o ataque não ocorre. Já que o remetente da mensagem final nunca pode ser certo de sua chegada, ele não correrá o risco de atacar. Pior ainda, o outro exército azul sabe disso, então também não atacará. Para ver a relevância do problema dos dois exércitos para a liberação de conexões, em vez do que para assuntos militares, apenas substitua "desconectar" por "ataque". Se nenhum dos lados estiver

Página 544

520

A CAMADA DE TRANSPORTE

INDIVÍDUO. 6

preparado para se desconectar até que esteja convencido de que o outro lado está preparado para desconectar também, a desconexão nunca acontecerá.

Na prática, podemos evitar esse dilema, dispensando a necessidade de acordo e levando o problema até o usuário do transporte, deixando cada lado independentemente decidir quando estiver pronto. Este é um problema mais fácil de resolver. A Figura 6-14 ilustra quatro cenários de liberação usando um handshake de três vias. Embora este protocolo seja não infalível, geralmente é adequado.

Na Figura 6-14 (a), vemos o caso normal em que um dos usuários envia um DR (SOLICITAÇÃO DE DESCONEXÃO) segmento para iniciar a liberação da conexão. Quando ele chega, o destinatário envia de volta um segmento DR e inicia um cronômetro, apenas no caso de seu

DR está perdido. Quando este DR chega, o remetente original envia de volta um segmento ACK e libera a conexão. Finalmente, quando o segmento ACK chega, o receptor também libera a conexão. Liberar uma conexão significa que o transporte entra

tity remove as informações sobre a conexão de sua tabela de atualmente aberta conexões e sinaliza o proprietário da conexão (o usuário de transporte) de alguma forma.

Esta ação é diferente de um usuário de transporte emitindo uma primitiva DISCONNECT .

Se o segmento ACK final for perdido, como mostrado na Fig. 6-14 (b), a situação é salvo pelo cronômetro. Quando o cronômetro expira, a conexão é encerrada de qualquer maneira.

Agora considere o caso da perda do segundo DR . O usuário iniciando o a desconexão não receberá a resposta esperada, atingirá o tempo limite e começará tudo de novo. Na Fig. 6-14 (c), vemos como isso funciona, assumindo que o segundo tempo nenhum segmento é perdido e todos os segmentos são entregues corretamente e no prazo.

Nosso último cenário, Fig. 6-14 (d), é o mesmo da Fig. 6-14 (c), exceto que agora nós assumir que todas as tentativas repetidas de retransmitir o DR também falham devido ao segmento perdido

mentos. Após N tentativas, o remetente simplesmente desiste e libera a conexão.

Enquanto isso, o receptor atinge o tempo limite e também sai.

Embora esse protocolo geralmente seja suficiente, em teoria ele pode falhar se o DR inicial e N retransmissões são todas perdidas. O remetente vai desistir e liberar a conexão, enquanto o outro lado não sabe nada sobre as tentativas de se desconectar e está ainda totalmente ativo. Esta situação resulta em uma conexão entreaberta.

Poderíamos ter evitado esse problema não permitindo que o remetente desistisse após N novas tentativas e forçando-o a continuar indefinidamente até obter uma resposta. No entanto, se

o outro lado pode atingir o tempo limite, o remetente continuará para sempre, porque nenhuma resposta será dada. Se não permitirmos que o lado receptor tempo limite, o protocolo trava na Figura 6.14 (d).

Uma maneira de matar conexões semiabertas é ter uma regra dizendo que se não houver segmentos chegaram por um certo número de segundos, a conexão é automaticamente desconectado. Dessa forma, se um lado se desconectar, o outro lado

detectar a falta de atividade e também desconectar. Esta regra também cuida do caso em que a conexão é interrompida (porque a rede não pode mais fornecer pacotes entre os hosts) sem nenhuma das extremidades se desconectar primeiro. Claro se esta regra é introduzida, é necessário que cada entidade de transporte tenha um temporizador que é interrompido e reiniciado sempre que um segmento é enviado. Se este cronômetro expirar, um

Página 545

SEC. 6,2

ELEMENTOS DE PROTOCOLOS DE TRANSPORTE

521

DR
ACK
ACK
Host 1
Host 2
DR
DR
Enviar DR
+ iniciar cronômetro
Enviar DR
+ iniciar cronômetro
Enviar ACK
Liberação
conexão
(Tempo esgotado)
liberação
conexão
(Tempo esgotado)
liberação
conexão
(N tempos limite)
liberação
conexão
(Tempo esgotado)
enviar DR
+ iniciar cronômetro
Liberação
conexão
DR
DR
Host 1
Host 2
DR
Enviar DR
+ iniciar cronômetro
Enviar DR &
iniciar cronômetro
Enviar DR &
iniciar cronômetro
Enviar DR &
iniciar cronômetro
Enviar ACK
Liberação
conexão
Liberação
conexão
DR
ACK
ACK
Host 1
Host 2
DR
Enviar DR
+ iniciar cronômetro
Enviar DR
+ iniciar cronômetro
Enviar ACK
Liberação
conexão
Perdido
Perdido
(Tempo esgotado)
enviar DR
+ iniciar cronômetro
DR
Host 1
Host 2
Enviar DR
+ iniciar cronômetro
Perdido
Perdido
(uma)

- (b)
- (c)
- (d)

Figura 6-14. Quatro cenários de protocolo para liberar uma conexão. (a) Normal caso de handshake de três vias. (b) ACK final perdido. (c) Resposta perdida. (d) Re-perda de response e perda de DRs subsequentes .

segmento fictício é transmitido, apenas para evitar que o outro lado se desconecte. Em por outro lado, se a regra de desconexão automática for usada e muitos segmentos falsos mentos consecutivos são perdidos em uma conexão ociosa, primeiro um lado, depois o outro será desconectado automaticamente.

Não entraremos mais em detalhes neste ponto, mas agora deve estar claro que liberar uma conexão sem perda de dados não é tão simples quanto parece à primeira vista. A lição aqui é que o usuário do transporte deve estar envolvido na decisão de quando

Página 546

522

A CAMADA DE TRANSPORTE INDIVÍDUO. 6

desconectar - o problema não pode ser resolvido de forma limpa pelas entidades de transporte deles-
eus. Para ver a importância da aplicação, considere que enquanto o TCP nor-
malmente faz um fechamento simétrico (com cada lado fechando independentemente sua metade do
conexão com um pacote FIN quando ele enviou seus dados), muitos servidores Web enviam
o cliente um pacote RST que provoca um fechamento abrupto da conexão que é mais
como um fechamento assimétrico. Isso funciona apenas porque o servidor Web conhece o padrão
tern de troca de dados. Primeiro, ele recebe uma solicitação do cliente, que é toda a
os dados que o cliente enviará e, em seguida, enviará uma resposta ao cliente. Quando a web
servidor terminou com sua resposta, todos os dados foram enviados em qualquer direção.
O servidor pode enviar ao cliente um aviso e encerrar abruptamente a conexão. Se o
cliente recebe este aviso, ele irá liberar seu estado de conexão imediatamente. Se o
o cliente não recebe o aviso, ele acabará percebendo que o servidor não está mais
er falando com ele e liberar o estado de conexão. Os dados foram com sucesso
transferido em qualquer caso.

6.2.4 Controle de Erro e Controle de Fluxo

Tendo examinado o estabelecimento e a liberação da conexão em alguns detalhes, vamos
agora ver como as conexões são gerenciadas enquanto estão em uso. Os principais problemas
são controle de erros e controle de fluxo. O controle de erros é garantir que os dados sejam entregues
com o nível de confiabilidade desejado, geralmente todos os dados são entregues
sem erros. O controle de fluxo está impedindo um transmissor rápido de ultrapassar um
receptor lento.

Ambos os problemas surgiram antes, quando estudamos o link de dados
camada. As soluções que são usadas na camada de transporte são os mesmos mecanismos
que estudamos no cap. 3. Como uma breve recapitulação:

1. Um frame carrega um código de detecção de erro (por exemplo, um CRC ou checksum)
que serve para verificar se a informação foi recebida corretamente.
2. Um quadro carrega um número de sequência para se identificar e é retransmitido
enviada pelo remetente até que receba uma confirmação de sucesso
recebimento do receptor. Isso é chamado de **ARQ (automático
Repita a solicitação)**.

3. Há um número máximo de frames que o remetente permitirá
ser excelente a qualquer momento, pausando se o receptor não for confirmado
quadros de impressão rápido o suficiente. Se este máximo for um pacote, o proto-
col é chamado de **parar e esperar**. Janelas maiores permitem pipelining e
melhorar o desempenho em links longos e rápidos.

4. O protocolo de **janela deslizante** combina esses recursos e também é
usado para suportar transferência de dados bidirecional.

Dado que esses mecanismos são usados em frames na camada de link, é natural
para se perguntar por que eles seriam usados em segmentos na camada de transporte também.

SEC. 6,2

ELEMENTOS DE PROTOCOLOS DE TRANSPORTE

523

No entanto, há pouca duplicação entre as camadas de enlace e transporte na prática tice. Mesmo que os mesmos mecanismos sejam usados, existem diferenças na função e grau.

Para uma diferença na função, considere a detecção de erros. A verificação da camada de link sum protege um quadro enquanto ele cruza um único link. A soma de verificação da camada de transporte

protege um segmento enquanto atravessa um caminho de rede inteiro. É um fim a fim verificar, o que não é o mesmo que verificar todos os links. Saltzer et al. (1984) descreve uma situação em que os pacotes foram corrompidos dentro de um roteador. A ligação somas de verificação de camada protegiam os pacotes apenas enquanto eles viajavam por um link, não

enquanto eles estavam dentro do roteador. Assim, os pacotes foram entregues incorretamente, mesmo

embora eles estivessem corretos de acordo com as verificações em cada link.

Este e outros exemplos levaram Saltzer et al. para articular o **argumento de ponta a ponta** **ment**. De acordo com este argumento, a verificação da camada de transporte que executa ponta a ponta

é essencial para a correção, e as verificações da camada de link não são essenciais, mas não menos valioso para melhorar o desempenho (uma vez que sem eles um pacote corrompido pode ser enviada ao longo de todo o caminho desnecessariamente).

Como uma diferença de grau, considere retransmissões e a janela deslizante protocolo. A maioria dos links sem fio, exceto links de satélite, pode ter apenas um único quadro pendente do remetente de cada vez. Ou seja, o produto de atraso de largura de banda pois o link é pequeno o suficiente para que nem mesmo um quadro inteiro possa ser armazenado dentro do

ligação. Nesse caso, um tamanho de janela pequeno é suficiente para um bom desempenho. Para exemplo, 802.11 usa um protocolo stop-and-wait, transmitindo ou retransmitindo cada quadro e esperando que ele seja reconhecido antes de passar para o próximo quadro.

Ter um tamanho de janela maior do que um quadro aumentaria a complexidade sem im- comprovando o desempenho. Para links com fio e de fibra óptica, como (comutado) Ether- backbones de rede ou ISP, a taxa de erro é baixa o suficiente para que as retransmissões da camada de link

pode ser omitido porque as retransmissões de ponta a ponta irão reparar o resíduo perda de quadros.

Por outro lado, muitas conexões TCP têm um produto de atraso de largura de banda que é muito maior do que um único segmento. Considere uma conexão enviando dados a- cruzar os EUA a 1 Mbps com um tempo de ida e volta de 100 mseg. Mesmo para este lento conexão, 200 Kbit de dados serão armazenados no receptor no tempo que leva para envie um segmento e receba uma confirmação. Para essas situações, um grande janela corrediça deve ser usada. Parar e esperar prejudicará o desempenho. Em nosso ex- amplo, limitaria o desempenho a um segmento a cada 200 mseg, ou 5 seg- mentos / s, não importa o quanto rápida a rede realmente seja.

Dado que os protocolos de transporte geralmente usam janelas deslizantes maiores, iremos observe a questão do armazenamento em buffer de dados com mais cuidado. Uma vez que um host pode ter muitos conexões, cada uma das quais tratada separadamente, pode precisar de uma quantidade substancial de buffer para as janelas deslizantes. Os buffers são necessários tanto no remetente e o receptor. Certamente, eles são necessários no remetente para manter todos os dados transmitidos mas segmentos ainda não reconhecidos. Eles são necessários lá porque estes seg- mentos podem ser perdidos e precisam ser retransmitidos.

No entanto, uma vez que o remetente está armazenando em buffer, o receptor pode ou não dedicar buffers específicos para conexões específicas, conforme achar adequado. O receptor pode, por exemplo, mantenha um único buffer pool compartilhado por todas as conexões. Quando um segmento entra, é feita uma tentativa de adquirir dinamicamente um novo buffer. Se um estiver disponível capaz, o segmento é aceito; caso contrário, ele é descartado. Uma vez que o remetente é pré-pareado para retransmitir segmentos perdidos pela rede, nenhum dano permanente é feito por tendo os segmentos de queda do receptor, embora alguns recursos sejam desperdiçados. O remetente continua tentando até obter uma confirmação.

A melhor compensação entre o buffer de origem e o buffer de destino depende no tipo de tráfego transportado pela conexão. Para tráfego intermitente de baixa largura de banda, como o produzido por um terminal interativo, é razoável não dedicar quaisquer buffers, mas sim adquiri-los dinamicamente em ambas as extremidades, contando com buffers-

remetente se os segmentos devem ser descartados ocasionalmente. No outro lado, para transferência de arquivos e outro tráfego de alta largura de banda, é melhor se o receptor dedica uma janela completa de buffers, para permitir que os dados fluam no máximo Rapidez. Esta é a estratégia que o TCP usa.

Ainda resta a questão de como organizar o buffer pool. Se mais segmentos são quase do mesmo tamanho, é natural organizar os buffers como um pool de buffers de tamanho idêntico, com um segmento por buffer, como na Figura 6-15 (a). No entanto, se houver grande variação no tamanho do segmento, de solicitações curtas de páginas da Web a grandes pacotes em transferências de arquivos ponto a ponto, um conjunto de buffers de tamanho fixo apresenta

problemas. Se o tamanho do buffer for escolhido para ser igual ao maior segmento possível, espaço será desperdiçado sempre que um segmento curto chegar. Se o tamanho do buffer for escolhido

senão ser menor que o tamanho máximo do segmento, vários buffers serão necessários para segmentos longos, com a complexidade do atendimento.

Outra abordagem para o problema do tamanho do buffer é usar buffers de tamanho variável, como na Figura 6-15 (b). A vantagem aqui é melhor utilização da memória, ao preço de gerenciamento de buffer mais complicado. Uma terceira possibilidade é dedicar um único buffer circular grande por conexão, como na Figura 6-15 (c). Este sistema é simples e elegante e não depende do tamanho dos segmentos, mas faz bom uso da memória somente quando as conexões estão muito carregadas.

Conforme as conexões são abertas e fechadas e conforme o padrão de tráfego muda, o emissor e o receptor precisam ajustar dinamicamente suas alocações de buffer. Consequentemente, o protocolo de transporte deve permitir que um host de envio solicite espaço de buffer

na outra extremidade. Buffers podem ser alocados por conexão, ou coletivamente, para todos as conexões em execução entre os dois hosts. Alternativamente, o receptor sabe a situação do buffer (mas sem saber o tráfego oferecido) poderia informar ao remetente "Eu reservei buffers X para você." Se o número de conexões abertas deve ser vinco, pode ser necessário que uma alocação seja reduzida, então o protocolo deve prever essa possibilidade.

Uma maneira razoavelmente geral de gerenciar a alocação dinâmica de buffer é desacoplar o armazenamento em buffer dos reconhecimentos, em contraste com a janela deslizante protocols do cap. 3. Gerenciamento de buffer dinâmico significa, com efeito, um tamanho variável

Segmento 3
 Segmento 4
 (uma)
 (b)
 (c)
 Não utilizado
 espaço

Figura 6-15. (a) Buffers de tamanho fixo em cadeia. (b) Buffers de tamanho variável em cadeia.
 (c) Um grande buffer circular por conexão.

janela. Inicialmente, o remetente solicita um certo número de buffers, com base em suas necessidades esperadas. O receptor então concede tantos destes quanto pode pagar. Cada vez que o remetente transmite um segmento, ele deve decrementar sua alocação, parando completamente quando a alocação chega a zero. O receptor separadamente pega carona confirmações e alocações de buffer no tráfego reverso. TCP usa este esquema, carregando alocações de buffer em um campo de cabeçalho denominado *tamanho da janela*.

A Figura 6-16 mostra um exemplo de como o gerenciamento dinâmico de janelas pode trabalhar em uma rede de datagramas com números de seqüência de 4 bits. Neste exemplo, dados fluxos em segmentos do host *A* para o host *B* e confirmações e buffer alocações fluem em segmentos na direção reversa. Inicialmente, *A* quer oito buffers, mas é concedido apenas quatro deles. Em seguida, ele envia três segmentos, dos quais o terceiro está perdido. O segmento 6 confirma o recebimento de todos os segmentos até e incluindo seqüência número 1, permitindo assim que *A* libere esses buffers e, além disso, informa *A* que tem permissão para enviar mais três segmentos começando além de 1 (ou seja, segmentos 2, 3 e 4). *A* sabe que já enviou o número 2, então pensa que pode enviar os segmentos 3 e 4, o que passa a fazer. Neste ponto é bloqueado e deve esperar por mais alocação de buffer. Retransmissão induzida por tempo limite (linha 9), no entanto, podem ocorrer enquanto bloqueadas, uma vez que usam buffers que têm já sido alocados. Na linha 10, *B* acusa o recebimento de todos os segmentos até e incluindo 4, mas se recusa a deixar *A* continuar. Tal situação é impossível com os protocolos de janela fixa do cap. 3. O próximo segmento de *B* para *A* aloca

526

A CAMADA DE TRANSPORTE INDIVÍDUO. 6

outro buffer e permite que *A* continue. Isso acontecerá quando *B* tiver buffer espaço, provavelmente porque o usuário de transporte aceitou mais dados de segmento.

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
<solicitar 8 buffers>
<ack = 15, buf = 4>
<seq = 0, dados = m0>
<seq = 1, dados = m1>
<seq = 2, dados = m2>
<ack = 1, buf = 3>
<seq = 3, dados = m3>
<seq = 4, dados = m4>
<seq = 2, dados = m2>
<ack = 4, buf = 0>
<ack = 4, buf = 1>
<ack = 4, buf = 2>
<seq = 5, dados = m5>
<seq = 6, dados = m6>
<ack = 6, buf = 0>
<ack = 6, buf = 4>
A quer 8 buffers

```

B concede mensagens 0-3 apenas
 A tem 3 buffers restantes agora
 A tem 2 buffers restantes agora
 Mensagem perdida, mas A pensa que ainda há 1
 B reconhece 0 e 1, permite 2-4
 A tem 1 buffer restante
 A tem 0 buffers restantes e deve parar
 A expira e retransmite
 Tudo confirmado, mas A ainda está bloqueado
 A agora pode enviar 5
 B encontrou um novo buffer em algum lugar
 A tem 1 buffer restante
 A agora está bloqueado novamente
 A ainda está bloqueado
 Potencial impasse
 UMA
 B
 mensagem
 Comentários

Figura 6-16. Alocação dinâmica de buffer. As setas mostram a direção de transmissão. Uma reticência (...) indica um segmento perdido.

Problemas com esquemas de alocação de buffer deste tipo podem surgir no datagrama redes se os segmentos de controle podem se perder - o que certamente pode acontecer. Veja na linha 16. *B* agora alocou mais buffers para *A*, mas o segmento de alocação foi perdido. Opa. Uma vez que os segmentos de controle não são sequenciados ou expirados, *A* é agora em um impasse. Para evitar essa situação, cada host deve enviar periodicamente o controle segmentos que fornecem a confirmação e o status do buffer em cada conexão. que forma, o impasse será quebrado, mais cedo ou mais tarde.

Até agora, assumimos tacitamente que o único limite imposto ao remetente taxa de dados é a quantidade de espaço de buffer disponível no receptor. Isso geralmente não é O caso. A memória já foi cara, mas os preços caíram drasticamente. Hosts pode estar equipado com memória suficiente para que a falta de buffers raramente, ou nunca, um problema, mesmo para conexões de longa distância. Claro, isso depende do buffer tamanho definido para ser grande o suficiente, o que nem sempre foi o caso do TCP (Zhang et al., 2002).

Quando o espaço do buffer não limita mais o fluxo máximo, outro gargalo aparecerá: a capacidade de suporte da rede. Se roteadores adjacentes podem ex-mude no máximo x pacotes / seg e existem k caminhos disjuntos entre um par de hosts, não há como esses hosts trocarem mais de kx segmentos / s, não verificar quanto espaço de buffer está disponível em cada extremidade. Se o remetente pressiona muito

SEC. 6,2

ELEMENTOS DE PROTOCOLOS DE TRANSPORTE

527

(ou seja, envia mais de kx segmentos / s), a rede ficará congestionada porque não será capaz de entregar segmentos tão rápido quanto eles estão entrando. O que é necessário é um mecanismo que limite as transmissões do remetente com base na capacidade de carga da rede, e não no buffer do receptor capacidade. Belsnes (1975) propôs o uso de um esquema de controle de fluxo de janela deslizante em que o remetente ajusta dinamicamente o tamanho da janela para corresponder ao da rede capacidade de carga. Isso significa que uma janela deslizante dinâmica pode implementar ambos controle de fluxo e controle de congestionamento. Se a rede pode lidar com segmentos c / s e o tempo de ida e volta (incluindo transmissão, propagação, enfileiramento, processing no receptor e retorno da confirmação) é r , a vitória do remetente dow deve ser cr . Com uma janela deste tamanho, o remetente normalmente opera com o pipeline está cheio. Qualquer pequena diminuição no desempenho da rede fará com que quadra. Como a capacidade de rede disponível para qualquer fluxo varia ao longo do tempo, o tamanho da janela deve ser ajustado com freqüência, para rastrear mudanças no transporte capacidade. Como veremos mais tarde, o TCP usa um esquema semelhante.

6.2.5 Multiplexação

Multiplexar ou compartilhar várias conversas por meio de conexões, circuitos virtuais cuits e links físicos desempenham um papel em várias camadas da arquitetura de rede.

Na camada de transporte, a necessidade de multiplexação pode surgir de várias maneiras. Por exemplo, se apenas um endereço de rede estiver disponível em um host, todas as condições de transporte

conexões nessa máquina têm que usá-lo. Quando um segmento chega, de alguma forma é preciso dizer a qual processo aplicá-lo. Esta situação, chamada de **multiplexação**, é mostrado na Fig. 6-17 (a). Nesta figura, quatro conexões de transporte distintas usam a mesma conexão de rede (por exemplo, endereço IP) para o host remoto.

A multiplexação também pode ser útil na camada de transporte por outro motivo. Supõe, por exemplo, que um host tem vários caminhos de rede que pode usar. Se um usuário precisa de mais largura de banda ou mais confiabilidade do que um dos caminhos de rede pode vide, uma saída é ter uma conexão que distribua o tráfego entre vários caminhos de rede em uma base round-robin, conforme indicado na Figura 6.17 (b). Este modus operandi é chamado de **multiplexação inversa**. Com k conexões de rede abertas, a largura de banda efetiva pode ser aumentada por um fator de k . Um exemplo de inverso a multiplexação é **SCTP (Stream Control Transmission Protocol)**, que pode ser executado uma conexão usando várias interfaces de rede. Em contraste, o TCP usa uma única rede ponto final de trabalho. A multiplexação inversa também é encontrada na camada de enlace, quando vários links de baixa taxa são usados em paralelo como um link de alta taxa.

6.2.6 Crash Recovery

Se hosts e roteadores estão sujeitos a travamentos ou as conexões são de longa duração (por exemplo, grandes downloads de software ou mídia), a recuperação dessas falhas torna-se um questão. Se a entidade de transporte estiver inteiramente dentro dos hosts, a recuperação da rede

Página 552

528

A CAMADA DE TRANSPORTE INDIVÍDUO. 6

Camada

4

3

2

1

Para o roteador

Linhas de roteador

Endereço de transporte

Rede

endereço

(uma)

(b)

Figura 6-17. (a) Multiplexação. (b) Multiplexação inversa.

e travar o roteador é simples. As entidades de transporte esperam segmentos perdidos o tempo todo e saber lidar com eles por meio de retransmissões.

Um problema mais problemático é como se recuperar de travamentos do host. Em particular lar, pode ser desejável que os clientes possam continuar trabalhando quando os servidores travar e reiniciar rapidamente. Para ilustrar a dificuldade, vamos supor que um host, o cliente, está enviando um arquivo longo para outro host, o servidor de arquivos, usando um simples

protocolo parar e esperar. A camada de transporte no servidor apenas passa a entrada segmentos para o usuário do transporte, um por um. No meio da transmissão, o o servidor trava. Quando ele volta, suas tabelas são reinicializadas, então ele não sabe exatamente onde estava.

Na tentativa de recuperar seu status anterior, o servidor pode enviar uma transmissão segmento para todos os outros hosts, anunciando que acabou de falhar e solicitando que seus clientes informam sobre o status de todas as conexões abertas. Cada cliente pode estar em um de dois estados: um segmento pendente, $S1$, ou nenhum segmento pendente, $S0$.

Com base apenas nessas informações de estado, o cliente deve decidir se retransmitirá o segmento mais recente.

À primeira vista, parece óbvio: o cliente deve retransmitir se e apenas se ele tem um segmento não reconhecido pendente (ou seja, está no estado $S1$) quando

aprende sobre o acidente. No entanto, uma inspeção mais minuciosa revela dificuldades com este abordagem ingênuas. Considere, por exemplo, a situação em que a transferência do servidor entidade esportiva primeiro envia uma confirmação e, em seguida, quando a confirmação foi enviado, escreve para o processo de candidatura. Escrevendo um segmento no exterior colocar stream e enviar uma confirmação são dois eventos distintos que não podem ser feito simultaneamente. Se ocorrer uma falha após o reconhecimento ter sido enviado, mas antes que a gravação seja totalmente concluída, o cliente receberá o

Página 553

SEC. 6,2

ELEMENTOS DE PROTOCOLOS DE TRANSPORTE

529

reconhecimento e, portanto, estar no estado $S0$ quando o anúncio de recuperação de falha chega. O cliente, portanto, não retransmitirá, (incorrectamente) pensando que o segmento chegou. Essa decisão do cliente leva a um segmento ausente.

Neste ponto, você pode estar pensando: " Esse problema pode ser resolvido facilmente. Todos você tem que fazer é reprogramar a entidade de transporte para primeiro fazer a gravação e, em seguida, enviar

o reconhecimento. " Tente novamente. Imagine que a gravação foi feita, mas o travamento ocorre antes que a confirmação possa ser enviada. O cliente estará no estado $S1$ e, portanto, retransmitir, levando a um segmento duplicado não detectado na saída stream para o processo de aplicativo do servidor.

Não importa como o cliente e o servidor são programados, sempre há situações ções em que o protocolo não recupera corretamente. O servidor pode ser programado de duas maneiras: confirme primeiro ou escreva primeiro. O cliente pode ser pró programado de uma das quatro maneiras: sempre retransmitir o último segmento, nunca retransmitir o último segmento, retransmitir apenas no estado $S0$ ou retransmitir apenas no estado $S1$. Isso dá oito combinações, mas como veremos, para cada combinação há

algum conjunto de eventos que faz com que o protocolo falhe.

Três eventos são possíveis no servidor: o envio de uma confirmação (A), gravação no processo de saída (W) e travamento (C). Os três eventos podem ocorrer em seis ordens diferentes: AC (W), AWC , C (AW), C (WA), WAC e WC (A), onde os parênteses são usados para indicar que nem A nem W podem seguir C (ou seja, assim que travar, travou). A Figura 6-18 mostra todas as oito combinações de estratégias de cliente e servidor e as sequências de eventos válidas para cada um. Aviso prévio que para cada estratégia há alguma sequência de eventos que faz com que o protocolo falhou. Por exemplo, se o cliente sempre retransmite, o evento AWC irá gerar

uma duplicata não detectada, embora os outros dois eventos funcionem corretamente.

Sempre retransmitir

Está bem

DUP

Está bem

PERDIDO

Está bem

PERDIDO

Está bem

DUP

PERDIDO

PERDIDO

Está bem

Está bem

Nunca retransmitir

Retransmitir em $S0$

Retransmitir em $S1$

AC (W)

Estratégia usada por
enviando hospedeiro

AWC

Primeiro ACK, depois escreva

Primeiro escreva, depois ACK

C (AW)

Está bem

DUP

DUP

PERDIDO

Está bem

Está bem

PERDIDO
DUP
Está bem
Está bem
Está bem
DUP
C (WA)
W AC
WC (A)
Está bem
= O protocolo funciona corretamente
DUP = protocolo gera uma mensagem duplicada
LOST = O protocolo perde uma mensagem
Estratégia usada pelo host de recebimento

Figura 6-18. Diferentes combinações de estratégias de cliente e servidor.

Página 554

530

A CAMADA DE TRANSPORTE INDIVÍDUO. 6

Tornar o protocolo mais elaborado não ajuda. Mesmo se o cliente e o servidor trocar vários segmentos antes que o servidor tente escrever, de modo que o cliente sabe exatamente o que está para acontecer, o cliente não tem como saber se uma falha ocorreu antes ou logo após a gravação. A conclusão é inevitável: sob nossas regras básicas de nenhum evento simultâneo, isto é, separado eventos acontecem um após o outro, não ao mesmo tempo - falha de host e recuperação não pode ser transparente para as camadas superiores.

Em termos mais gerais, este resultado pode ser reafirmado como "recuperação de um a queda da camada N só pode ser feita pela camada $N+1$," e apenas se a camada superior retém informações de status suficientes para reconstruir onde estava antes do problema ocorreu. Isso é consistente com o caso mencionado acima de que o transporte camada pode se recuperar de falhas na camada de rede, desde que cada extremidade de um a conexão mantém o controle de onde está.

Este problema nos leva à questão do que é chamado de reconhecimento de ponta a ponta edge-to-edge realmente significa. Em princípio, o protocolo de transporte é ponta a ponta e não acorrentados como as camadas inferiores. Agora considere o caso de um usuário inserir solicitações para transações em um banco de dados remoto. Suponha que o transporte remoto entity é programado para primeiro passar segmentos para a próxima camada e depois reconhecer borda. Mesmo neste caso, o recebimento de uma confirmação de volta no usuário máquina não significa necessariamente que o host remoto permaneceu ativo tempo suficiente para realmente atualizar o banco de dados. Uma confirmação verdadeiramente ponta a ponta, cujo recibo significa que o trabalho foi realmente feito e a falta dele significa que não foi, é provavelmente impossível de alcançar. Este ponto é discutido em mais detalhes por Saltzer et al. (1984).

6.3 CONTROLE DE CONGESTÃO

Se as entidades de transporte em muitas máquinas enviarem muitos pacotes para a rede trabalhar muito rápido, a rede ficará congestionada, com desempenho degradado porque os pacotes são atrasados e perdidos. Controlar o congestionamento para evitar este problema é

a responsabilidade combinada das camadas de rede e transporte. Congestionamento occurs em roteadores, por isso é detectado na camada de rede. No entanto, o congestionamento é ultimamente causado pelo tráfego enviado para a rede pela camada de transporte. O único efeito maneira eficaz de controlar o congestionamento é os protocolos de transporte enviarem pacotes na rede mais lentamente.

No cap. 5, estudamos mecanismos de controle de congestionamento na camada de rede.

Nesta seção, estudaremos a outra metade do problema, o controle de congestionamento mecanismos na camada de transporte. Depois de descrever os objetivos do congestionamento control, vamos descrever como os hosts podem regular a taxa de envio de pacotes na rede. A Internet depende muito da camada de transporte para o congestionamento controle e algoritmos específicos são integrados ao TCP e a outros protocolos.

SEC. 6,3
CONTROLE DE CONGESTÃO

531

6.3.1 Alocação de largura de banda desejável

Antes de descrevermos como regular o tráfego, devemos entender o que somos tentando alcançar executando um algoritmo de controle de congestionamento. Ou seja, devemos especificar o estado em que um bom algoritmo de controle de congestionamento operará a rede. O objetivo é mais do que simplesmente evitar congestionamentos. É encontrar um bom allocalização da largura de banda para as entidades de transporte que estão usando a rede. Um bem alocação vai entregar um bom desempenho porque usa toda a banda disponível largura, mas evita congestionamento, será justo em entidades de transporte concorrentes, e ele rastreará rapidamente as mudanças nas demandas de tráfego. Faremos cada uma dessas critérios mais precisos por sua vez.

Eficiência e Poder

Uma alocação eficiente de largura de banda entre as entidades de transporte usará todos os a capacidade de rede disponível. No entanto, não é certo pensar que se há um link de 100 Mbps, cinco entidades de transporte devem obter 20 Mbps cada. Eles geralmente deve obter menos de 20 Mbps para um bom desempenho. A razão é que o tráfego costuma ser intermitente. Lembre-se de que na seção 5.3 descrevemos o **goodput** (ou taxa de pacotes úteis que chegam ao receptor) em função da carga oferecida. Isto curva e uma curva correspondente para o atraso em função da carga oferecida são dado na Figura 6-19.

Capacidade
(uma)
Carga oferecida (pacotes / s)
Congestionamento
colapso
Carga oferecida (pacotes / s)
Goodput
(pacotes / s)
Desejado
resposta
Demora
(segundos)
(b)
Início de
congestionamento

Figura 6-19. (a) Goodput e (b) atraso em função da carga oferecida.

À medida que a carga aumenta na Fig. 6-19 (a) goodput inicialmente aumenta ao mesmo taxa, mas conforme a carga se aproxima da capacidade, goodput aumenta mais gradualmente. Isto queda é porque rajadas de tráfego podem ocasionalmente aumentar e causar perdas em buffers dentro da rede. Se o protocolo de transporte for mal projetado e retransmite pacotes que foram atrasados, mas não perdidos, a rede pode entrar colapso do congestionamento. Nesse estado, os remetentes estão enviando pacotes furiosamente, mas cada vez mais pouco trabalho útil está sendo realizado.

532

A CAMADA DE TRANSPORTE
INDIVÍDUO. 6

O atraso correspondente é dado na Fig. 6-19 (b) Inicialmente, o atraso é fixo, representando o atraso de propagação pela rede. Conforme a carga se aproxima do capacidade, o atraso aumenta, lentamente no início e depois muito mais rapidamente. Isso é de novo por causa de picos de tráfego que tendem a aumentar com cargas elevadas. O atraso não pode realmente vêm para o infinito, exceto em um modelo no qual os roteadores têm buffers infinitos. Em vez disso, os pacotes serão perdidos após experimentar o atraso máximo de buffer. Para goodput e atraso, o desempenho começa a degradar no início de congestionamento. Intuitivamente, obteremos o melhor desempenho da rede se alocamos largura de banda até que o atraso comece a aumentar rapidamente. Este ponto é ser-

reduza a capacidade. Para identificá-lo, Kleinrock (1979) propôs a métrica do **poder**,

Onde

$$\text{potência} = \frac{\text{demora}}{\text{carga}}$$

A potência inicialmente aumentará com a carga oferecida, pois o atraso permanece pequeno e aproximadamente

constante, mas atingirá um máximo e diminuirá conforme o atraso aumenta rapidamente. A carga com

a maior potência representa uma carga eficiente para a entidade de transporte colocar a rede.

Justiça Max-Min

Na discussão anterior, não falamos sobre como dividir a largura de banda entre diferentes remetentes de transporte. Parece uma pergunta simples para resposta - dê a todos os remetentes uma fração igual da largura de banda - mas envolve várias considerações.

Talvez a primeira consideração seja perguntar o que esse problema tem a ver com controle de gestão. Afinal, se a rede dá a um remetente alguma quantidade de largura de banda para usar, o remetente deve apenas usar essa quantidade de largura de banda. No entanto, muitas vezes é o

caso as redes não tenham uma reserva estrita de largura de banda para cada fluxo ou conexão. Eles podem para alguns fluxos se a qualidade do serviço for suportada, mas muitos as conexões procurarão usar qualquer largura de banda disponível ou serão agrupadas ela pela rede sob uma atribuição comum. Por exemplo, a diferenciação da IETF serviços ed separam o tráfego em duas classes e as conexões competem por banda largura dentro de cada classe. Os roteadores IP costumam ter todas as conexões competindo pelo mesmo largura de banda. Nesta situação, é o mecanismo de controle de congestionamento que é alocar largura de banda para as conexões concorrentes.

Uma segunda consideração é o que uma porção justa significa para fluxos em uma rede. isto é bastante simples se N fluxos usam um único link, caso em que todos eles podem ter $1/N$ da largura de banda (embora a eficiência dite que eles usem um pouco menos se o tráfego é intermitente). Mas o que acontece se os fluxos forem diferentes, mas sobrepostos, caminhos de rede? Por exemplo, um fluxo pode cruzar três links, e os outros fluxos pode cruzar um link. O fluxo de três links consome mais recursos da rede. isto pode ser mais justo, em certo sentido, fornecer menos largura de banda do que os fluxos de um link. isto

Página 557

SEC. 6,3

CONTROLE DE CONGESTÃO

533

deve certamente ser possível suportar mais fluxos de um link, reduzindo a largura do fluxo de três elos. Este ponto demonstra uma tensão inerente entre justiça e eficiência.

No entanto, vamos adotar uma noção de justiça que não depende do comprimento do caminho da rede. Mesmo com este modelo simples, dando às conexões um fração igual de largura de banda é um pouco complicado porque conexões diferentes tomará caminhos diferentes através da rede e esses caminhos terão capacidades diferentes. Neste caso, é possível que um fluxo seja obstruído em um link downstream e tomar uma porção menor de um link upstream do que outros fluxos; reduzir a largura de banda dos outros fluxos os tornaria mais lentos, mas não ajudar o fluxo obstruído.

A forma de justiça que muitas vezes é desejada para o uso da rede é **máximo-mínimo justo** **ness**. Uma alocação é máxima-mínima justa se a largura de banda dada a um fluxo não pode ser aumentou sem diminuir a largura de banda dada a outro fluxo com uma alocação que não é maior. Ou seja, aumentar a largura de banda de um fluxo só fará a situação é pior para fluxos menos favorecidos.

Vejamos um exemplo. Uma alocação justa máx-mín é mostrada para uma rede com quatro fluxos, A , B , C e D , na Fig. 6-20. Cada um dos links entre os roteadores tem a mesma capacidade, considerada como 1 unidade, embora no caso geral os links terão capacidades diferentes. Três fluxos competem pelo link inferior esquerdo entre os roteadores $R4$ e $R5$. Cada um desses fluxos, portanto, obtém $1/3$ do link. O restante o fluxo, A , compete com B no link de $R2$ para $R3$. Uma vez que B tem uma alocação de $1/3$, A obtém os $2/3$ restantes do link. Observe que todos os outros links têm capacidade extra. No entanto, esta capacidade não pode ser dada a qualquer um dos fluxos sem diminuir a capacidade de outro, menor fluxo. Por exemplo, se mais da banda largura na ligação entre $R2$ e $R3$ é dada ao fluxo B , haverá menos para o fluxo Um . Isso é razoável porque o fluxo A já tem mais largura de banda. No entanto, a velocidade do fluxo C ou D (ou ambos) deve ser diminuída para dar mais largura de banda para B , e esses fluxos terão menos banda do que B . Assim, a alocação é max-min justo.

```

1/3
R1
R2
D
C
B
UMA
1/3
2/3
1/3
1/3
1/3
1/3
D
C
B
UMA
R4
R3
R6
R5
2/3
1/3

```

Figura 6-20. Alocação de largura de banda máxima-mínima para quatro fluxos.

As alocações máximas-mínimas podem ser calculadas, dado um conhecimento global da rede trabalhos. Uma maneira intuitiva de pensar sobre eles é imaginar que a taxa de todos os

Página 558

534

A CAMADA DE TRANSPORTE INDIVÍDUO. 6

os fluxos começam em zero e aumentam lentamente. Quando a taxa atinge um gargalo para qualquer fluxo, então esse fluxo para de aumentar. Todos os outros fluxos continuam a aumentar, compartilhando igualmente a capacidade disponível, até que também atinjam seus respectivos bottlenecks.

Uma terceira consideração é o nível sobre o qual considerar a justiça. Uma rede pode ser justo no nível de conexões, conexões entre um par de hosts, ou todas as conexões por host. Examinamos esse problema quando discutimos o WFQ (Weighted Fair Queuing) na seção 5.4 e concluiu que cada uma dessas definições tem seus problemas. Por exemplo, definir justiça por host significa que um servidor ocupado não se sairá melhor do que um telefone celular, ao definir justiça por conexão incentiva os hosts a abrir mais conexões. Dado que não há uma resposta clara, justiça é frequentemente considerada por conexão, mas justiça precisa geralmente não é uma preocupação. É mais importante na prática que nenhuma conexão seja privada de banda largura do que todas as conexões obtêm precisamente a mesma quantidade de largura de banda. No de fato, com o TCP é possível abrir várias conexões e competir por banda largura de forma mais agressiva. Essa tática é usada por aplicativos que consomem muita largura de banda

como o BitTorrent para compartilhamento de arquivos ponto a ponto.

Convergência

Um critério final é que o algoritmo de controle de congestionamento converge rapidamente para um

alocação justa e eficiente de largura de banda. A discussão da operação desejável ponto inicial acima pressupõe um ambiente de rede estático. No entanto, as conexões são sempre indo e vindo em uma rede, e a largura de banda necessária para um determinado con conexão também irá variar com o tempo, por exemplo, quando um usuário navega em páginas da Web e ocasionalmente baixa vídeos grandes.

Por causa da variação na demanda, o ponto operacional ideal para a rede varia com o tempo. Um bom algoritmo de controle de congestionamento deve convergir rapidamente para

o ponto operacional ideal e deve rastreá-lo à medida que ele muda com o tempo. E se a convergência é muito lenta, o algoritmo nunca estará perto da mudança de op ponto de referência. Se o algoritmo não for estável, ele pode falhar em convergir para a direita apontar em alguns casos, ou mesmo oscilar em torno do ponto certo.

Um exemplo de uma alocação de largura de banda que muda com o tempo e converge rapidamente é mostrado na Fig. 6-21. Inicialmente, o fluxo 1 possui toda a largura de banda. Um segundo-

depois, o fluxo 2 começa. Ele também precisa de largura de banda. A alocação rapidamente mudanças para dar a cada um desses fluxos metade da largura de banda. Em 4 segundos, um terceiro fluxo

junta-se. No entanto, esse fluxo usa apenas 20% da largura de banda, o que é menor que seu parte justa (que é um terço). Os fluxos 1 e 2 se ajustam rapidamente, dividindo o disponível largura de banda para cada um ter 40% da largura de banda. Aos 9 segundos, o segundo fluxo folhas, e o terceiro fluxo permanece inalterado. O primeiro fluxo captura rapidamente 80% da largura de banda. Em todos os momentos, a largura de banda total alocada é de aproximadamente 100%, para que a rede seja totalmente utilizada e os fluxos concorrentes recebam tratamento igual (mas não precisa usar mais largura de banda do que o necessário).

Página 559

SEC. 6,3

CONTROLE DE CONGESTÃO

535

Fluxo 1
0,5
Tempo (segundos)
Largura de banda
alocação
0
1
1
4
9
Fluxo 3
Fluxo 2 paradas
Fluxo 2 começa

Figura 6-21. Alterar a alocação de largura de banda ao longo do tempo.

6.3.2 Regulando a taxa de envio

Agora é a hora do prato principal. Como regulamos as taxas de envio para obter uma alocação de largura de banda desejável? A taxa de envio pode ser limitada por dois fatores. O primeiro é o controle de fluxo, no caso de haver buffer insuficiente em o receptor. O segundo é o congestionamento, no caso de haver capacidade insuficiente na rede. Na Figura 6-22, vemos esse problema ilustrado hidráulicamente. Na Na Fig. 6-22 (a), vemos um tubo grosso levando a um receptor de pequena capacidade. Isto é um situação limitada de controle de fluxo. Contanto que o remetente não envie mais água do que o balde pode conter, nenhuma água será perdida. Na Fig. 6-22 (b), a limitação O fator não é a capacidade do balde, mas a capacidade de carga interna da rede. Se entrar muita água muito rápido, ela voltará e parte será perdida (neste caso, transbordando o funil).

Esses casos podem parecer semelhantes para o remetente, pois transmitem causas muito rápidas pacotes a serem perdidos. No entanto, eles têm causas diferentes e exigem soluções diferentes ções. Já falamos sobre uma solução de controle de fluxo com um tamanho variável

janela. Agora vamos considerar uma solução de controle de congestionamento. Desde qualquer um dos

esses problemas podem ocorrer, o protocolo de transporte geralmente precisará executar ambos soluções e diminua a velocidade se algum dos problemas ocorrer.

A maneira como um protocolo de transporte deve regular a taxa de envio depende de a forma de feedback retornado pela rede. Diferentes camadas de rede podem retornar diferentes tipos de feedback. O feedback pode ser explícito ou implícito e pode ser preciso ou impreciso.

Um exemplo de design explícito e preciso é quando os roteadores informam às fontes que taxa na qual eles podem enviar. Projetos na literatura, como XCP (eXplicit Congestion Control) operam dessa maneira (Katabi et al., 2002). Um explícito, impreciso design é o uso de ECN (Explicit Congestion Notification) com TCP. Nesse design, os roteadores definem bits nos pacotes que sofrem congestionamento para avisar os remetentes

desacelerar, mas eles não dizem o quanto desacelerar.

Página 560

536

A CAMADA DE TRANSPORTE INDIVÍDUO. 6

Transmissão

ajuste de taxa

Transmissão

rede

interno

congestionamento

Pequena capacidade

receptor

Grande capacidade

receptor

(uma)

(b)

Figura 6-22. (a) Uma rede rápida alimentando um receptor de baixa capacidade. (b) Um lento rede alimentando um receptor de alta capacidade.

Em outros projetos, não há sinal explícito. FAST TCP mede a rodada retardo de viagem e usa essa métrica como um sinal para evitar congestionamento (Wei et al., 2006). Finalmente, na forma de controle de congestionamento mais prevalente na Internet hoje, TCP com roteadores drop-tail ou RED, a perda de pacotes é inferida e usada para sinalizar que a rede ficou congestionada. Existem muitas variantes desta forma de TCP, incluindo CUBIC TCP, que é usado no Linux (Ha et al., 2008). Combinações também são possíveis. Por exemplo, o Windows inclui TCP composto que usa ambos perda e atraso de pacotes como sinais de feedback (Tan et al., 2006). Esses designs são resumido na Figura 6-23.

Se um sinal explícito e preciso for dado, a entidade de transporte pode usar esse sinal para ajustar sua taxa ao novo ponto de operação. Por exemplo, se o XCP diz aos remetentes a taxa a ser usada, os remetentes podem simplesmente usar essa taxa. Nos outros casos, no entanto, alguma suposição está envolvida. Na ausência de um sinal de congestionamento, os remetentes devem diminuir suas taxas. Quando um sinal de congestionamento é dado, os remetentes devem diminuir suas taxas. A forma como as taxas são aumentadas ou diminuídas é dada por uma **lei de controle**. Essas leis têm um grande efeito no desempenho.

Página 561

SEC. 6,3

CONTROLE DE CONGESTÃO

537

Protocolo

Sinal

Explícito? Preciso?

XCP

Taxa de uso

sim

sim

TCP com ECN	
Aviso de congestionamento	sim
	Não
FAST TCP	
Atraso de ponta a ponta	Não
	sim
TCP composto	
Perda de pacotes e atraso de ponta a ponta	Não
	sim
CUBIC TCP	
Perda de pacote	Não
	Não
TCP	
Perda de pacote	Não
	Não

Figura 6-23. Sinais de alguns protocolos de controle de congestionamento.

Chiu e Jain (1989) estudaram o caso de feedback de congestionamento binário e concluiu que AIMD (Additive Aumento Multiplicativo Diminuição) é o apropriado lei de controle para chegar a um ponto operacional eficiente e justo. Para discutir isso caso, eles construíram um argumento gráfico para o caso simples de dois conexões competindo pela largura de banda de um único link. O gráfico da Fig. 6-24 mostra a largura de banda alocada para o usuário 1 no eixo x ao usuário 2 no eixo y. Quando a alocação for justa, ambos os usuários receberão a mesma quantidade de largura de banda. Isso é mostrado pela linha pontilhada de justiça. Quando as alocações somam 100%, o capacidade do link, a alocação é eficiente. Isso é mostrado pelo efeito pontilhado linha de ciência. Um sinal de congestionamento é fornecido pela rede para ambos os usuários quando o soma de suas alocações cruza esta linha. A interseção dessas linhas é o ponto operacional desejado, quando ambos os usuários têm a mesma largura de banda e toda a rede largura de banda de trabalho é usada.

Aumento de aditivo	
e diminuir	
Largura de banda do usuário 1	
Linha de justiça	
Linha de eficiência	
Ponto ótimo	
Do utilizador	
2	
de	
largura de banda	
0	
Aumento multiplicativo	
e diminuir	
100%	
100%	

Figura 6-24. Ajustes de largura de banda aditivos e multiplicativos.

Considere o que acontece a partir de alguma alocação inicial se o usuário 1 e o usuário 2 aumentar adicionamente suas respectivas larguras de banda ao longo do tempo. Por exemplo, os usuários cada um pode aumentar sua taxa de envio em 1 Mbps a cada segundo. Eventualmente, o

ponto de operação cruza a linha de eficiência e ambos os usuários recebem um sinal de congestionamento. Nesse estágio, eles devem reduzir suas alocações. Contudo, uma diminuição de aditivo simplesmente faria com que oscilar ao longo de uma linha de aditivo. Essa situação é mostrada na Figura 6-24. O comportamento manterá o ponto operacional perto de eficiente, mas não será necessariamente justo.

Da mesma forma, considere o caso em que ambos os usuários aumentam multiplicativamente seus largura de banda ao longo do tempo até receberem um sinal de congestionamento. Por exemplo, os usuários

pode aumentar sua taxa de envio em 10% a cada segundo. Se eles então se multipliquem diminuir significativamente suas taxas de envio, o ponto operacional dos usuários simplesmente oscilar ao longo de uma linha multiplicativa. Esse comportamento também é mostrado na Figura 6.24.

A linha multiplicativa tem uma inclinação diferente da linha aditiva. (Aponta para o origem, enquanto a linha aditiva tem um ângulo de 45 graus.) Mas, caso contrário, não Melhor. Em nenhum dos casos os usuários convergirão para as taxas de envio ideais que são justo e eficiente.

Agora, considere o caso de os usuários aumentarem sua largura de banda adicionalmente locais e, em seguida, diminua-os multiplicativamente quando o congestionamento for. Esse comportamento é a lei de controle do AIMD e é mostrado na Figura 6.25. Pode ser visto que o caminho traçado por este comportamento converge para o ponto ideal que é justo e eficiente. Essa convergência acontece independentemente do início ponto, tornando AIMD amplamente útil. Pelo mesmo argumento, o único outro com binação, aumento multiplicativo e diminuição aditiva, divergiriam da ponto ideal.

Começar
Largura de banda do usuário 1 100%
Linha de justiça
Linha de eficiência
Ponto ótimo
Do utilizador
2's
largura de banda
= Aumento de aditivo
(até 45)
= Diminuição multiplicativa
(linha aponta para a origem)
Lenda:
100%
0
0

Figura 6-25. Lei de controle de redução multiplicativa de aumento aditivo (AIMD).

AIMD é a lei de controle que é usada pelo TCP, com base neste argumento e um outro argumento de estabilidade (que é fácil conduzir a rede ao congestionamento e difícil de recuperar, então a política de aumento deve ser suave e a política de diminuição é agressivo). Não é muito justo, já que as conexões TCP ajustam sua janela tamanho por uma determinada quantidade a cada tempo de ida e volta. Diferentes conexões terão diferentes tempos de ida e volta. Isso leva a um viés em que as conexões com hosts mais próximos recebem mais largura de banda do que conexões com hosts distantes, se todo o resto for igual.

Página 563

SEC. 6,3

CONTROLE DE CONGESTÃO

539

Na seção 6.5, iremos descrever em detalhes como o TCP implementa um controle AIMD lei para ajustar a taxa de envio e fornecer controle de congestionamento. Esta tarefa é mais difícil do que parece porque as taxas são medidas ao longo de algum intervalo e tráfego está com rajadas. Em vez de ajustar a taxa diretamente, uma estratégia que costuma ser usada em a prática é ajustar o tamanho de uma janela deslizante. O TCP usa essa estratégia. Se o o tamanho da janela é W e o tempo de ida e volta é RTT , a taxa equivalente é W / RTT . Essa estratégia é fácil de combinar com o controle de fluxo, que já usa uma janela, e tem a vantagem de que o remetente acompanha os pacotes usando confirmações e portanto, fica lento em um RTT se parar de receber relatórios de que os pacotes estão saindo a rede.

Como questão final, pode haver muitos protocolos de transporte diferentes que enviam tráfego na rede. O que acontecerá se os diferentes protocolos competirem com

diferentes leis de controle para evitar congestionamentos? Alocações desiguais de largura de banda, isto é o que. Uma vez que o TCP é a forma dominante de controle de congestionamento na Internet, há é a pressão significativa da comunidade para que novos protocolos de transporte sejam projetados de forma que eles competem de forma justa com ele. Os primeiros protocolos de streaming media causaram problemas lems reduzindo excessivamente a taxa de transferência de TCP porque não competiam bastante. Isso levou à noção de controle de congestionamento **compatível** com TCP em que o TCP e os protocolos de transporte não TCP podem ser misturados livremente sem efeitos nocivos (Floyd et al., 2000).

6.3.3 Problemas sem fio

Os protocolos de transporte, como TCP, que implementam o controle de congestionamento, devem ser independente da rede subjacente e das tecnologias da camada de link. Isso é um bom teoria, mas na prática existem problemas com redes sem fio. O principal problema é que a perda de pacotes é frequentemente usada como um sinal de congestionamento, incluindo pelo TCP, como temos

apenas discutido. As redes sem fio perdem pacotes o tempo todo devido ao erro de transmissão rors.

Com a lei de controle AIMD, alto rendimento requer níveis muito pequenos de perda de pacotes. As análises de Padhye et al. (1998) mostram que o rendimento aumenta à medida que

a raiz quadrada inversa da taxa de perda de pacotes. O que isso significa na prática é que a taxa de perda para conexões TCP rápidas é muito pequena; 1% é uma taxa de perda moderada, e quando a taxa de perda chega a 10%, a conexão foi efetivamente interrompida trabalhando. No entanto, para redes sem fio, como 802.11 LANs, taxas de perda de quadros de pelo menos 10% são comuns. Esta diferença significa que, sem medidas de proteção dados, esquemas de controle de congestionamento que usam a perda de pacotes como um sinal serão desnecessários

limitar necessariamente as conexões que funcionam em links sem fio a taxas muito baixas.

Para funcionar bem, as únicas perdas de pacotes que o algoritmo de controle de congestionamento deve observar são perdas devido a largura de banda insuficiente, não perdas devido a trans- erros de missão. Uma solução para este problema é mascarar as perdas sem fio por usando retransmissões pelo link sem fio. Por exemplo, 802.11 usa um stop- e espere o protocolo para entregar cada quadro, repetindo as transmissões várias vezes se

Página 564

540

A CAMADA DE TRANSPORTE
INDIVÍDUO. 6

precisa ser antes de relatar uma perda de pacote para a camada superior. No caso normal, cada pacote é entregue apesar dos erros de transmissão transitórios que não são visíveis para o camadas superiores.

A Fig. 6-26 mostra um caminho com um link com e sem fio para o qual o mascaramento estratégia é usada. Existem dois aspectos a serem observados. Primeiro, o remetente não precisa necessariamente saber que o caminho inclui um link sem fio, uma vez que tudo o que vê é o link com fio

ao qual está anexado. Os caminhos da Internet são heterogêneos e não há método para o remetente dizer que tipo de links compõem o caminho. Este compli- cata o problema de controle de congestionamento, pois não há uma maneira fácil de usar um protocolo

para links sem fio e outro protocolo para links com fio.

Link com fio
Remetente
Receptor

Transporte com controle de congestionamento de ponta a ponta (perda = congestionamento)

Retransmissão da camada de link
(perda = erro de transmissão)

Link sem fio

Figura 6-26. Controle de congestionamento em um caminho com um link sem fio.

O segundo aspecto é um quebra-cabeça. A figura mostra dois mecanismos que são impulsionados pela perda: retransmissões do frame da camada de link e congestionamento da camada de transporte

ao controle. O quebra-cabeça é como esses dois mecanismos podem coexistir sem obter confuso. Afinal, uma perda deve fazer com que apenas um mecanismo entre em ação porque é um erro de transmissão ou um sinal de congestionamento. Não pode ser ambos. E se ambos os mecanismos entram em ação (retransmitindo o quadro e desacelerando o taxa de envio), então estamos de volta ao problema original dos transportes que vão longe muito lentamente em links sem fio. Considere este quebra-cabeça por um momento e veja se você pode resolver isso.

A solução é que os dois mecanismos atuam em escalas de tempo diferentes. Ligação as retransmissões da camada acontecem na ordem de microssegundos a milissegundos por links sem fio, como 802.11. Temporizadores de perda em protocolos de transporte disparam no pedido

de milissegundos para segundos. A diferença é de três ordens de magnitude. Este algoritmo links sem fio para detectar perdas de quadros e retransmite quadros para reparar trans-erros de missão muito antes de a perda de pacotes ser inferida pela entidade de transporte.

A estratégia de mascaramento é suficiente para permitir que a maioria dos protocolos de transporte funcione bem

na maioria dos links sem fio. No entanto, nem sempre é uma solução adequada. Alguns os links sem fio têm longos tempos de ida e volta, como satélites. Para esses links outras técnicas devem ser usadas para mascarar a perda, como FEC (Forward Error Correction), ou o protocolo de transporte deve usar um sinal de não perda para controle de congestionamento.

Página 565

SEC. 6,3
CONTROLE DE CONGESTÃO

541

Um segundo problema com o controle de congestionamento em links sem fio é a capacidade variável

ty. Ou seja, a capacidade de um link sem fio muda com o tempo, às vezes abruptamente, conforme os nós se movem e a relação sinal-ruído varia com a mudança de canal dições. Isso é diferente de links com fio, cuja capacidade é fixa. O protocolo de transporte deve se adaptar à capacidade de mudança dos links sem fio, caso contrário, gerar a rede ou deixar de usar a capacidade disponível.

Uma solução possível para esse problema é simplesmente não se preocupar com isso. Isto estratégia é viável porque algoritmos de controle de congestionamento já devem lidar com o caso de novos usuários entrando na rede ou usuários existentes alterando seu envio cotações. Mesmo que a capacidade dos links com fio seja fixa, a mudança de comportamento de outros usuários se apresentam como uma variabilidade na largura de banda que está disponível para um

determinado usuário. Assim, é possível simplesmente executar o TCP em um caminho com um fio 802.11

menos link e obter um desempenho razoável.

No entanto, quando há muita variabilidade sem fio, os protocolos de transporte decaídos para links com fio podem ter problemas para se manter atualizados e apresentar baixo desempenho.

A solução nesse caso é um protocolo de transporte projetado para links sem fio.

Um cenário particularmente desafiador é uma rede mesh sem fio na qual vários, os links sem fio interferentes devem ser cruzados, as rotas alteradas devido à mobilidade e há muitas perdas. Pesquisas nesta área estão em andamento. Veja Li et al. (2009) para um exemplo de projeto de protocolo de transporte sem fio.

6.4 OS PROTOCOLOS DE TRANSPORTE DA INTERNET: UDP

A Internet tem dois protocolos principais na camada de transporte, um sem conexão

protocolo e um orientado a conexão. Os protocolos se complementam. O protocolo sem conexão é UDP. Não faz quase nada além de enviar pacote conjuntos entre aplicativos, permitindo que os aplicativos construam seus próprios protocolos no topo como necessário. O protocolo orientado a conexão é o TCP. Faz quase tudo. Isto faz conexões e adiciona confiabilidade com retransmissões, junto com o fluxo de controle e controle de congestionamento, tudo em nome dos aplicativos que o utilizam. Nas seções a seguir, estudaremos UDP e TCP. Vamos começar com UDP porque é o mais simples. Também examinaremos dois usos do UDP. Uma vez que UDP é um protocolo de camada de transporte que normalmente é executado no sistema operacional e protocolos que usam UDP normalmente executado no espaço do usuário, esses usos podem ser considerados aplicativos. No entanto, as técnicas que eles usam são úteis para muitas aplicações e são melhor considerado pertencer a um serviço de transporte, por isso iremos abordá-los aqui.

6.4.1 Introdução ao UDP

O pacote de protocolos da Internet suporta um protocolo de transporte sem conexão chamado **UDP (protocolo de datagrama do usuário)**. UDP fornece uma maneira para os aplicativos enviarem datagramas IP encapsulados sem a necessidade de estabelecer uma conexão. UDP é descrito no RFC 768.

Página 566

542

A CAMADA DE TRANSPORTE INDIVÍDUO. 6

UDP transmite **segmentos que** consistem em um cabeçalho de 8 bytes seguido pelo cargo. O cabeçalho é mostrado na Figura 6-27. As duas **portas** servem para identificar o fim pontos nas máquinas de origem e destino. Quando um pacote UDP chega, sua carga útil é entregue ao processo conectado à porta de destino. Este anexo ocorre quando o primitivo BIND ou algo semelhante é usado, como vimos em Figura 6-6 para TCP (o processo de vinculação é o mesmo para UDP). Pense nas portas como caixas de correio que os aplicativos podem alugar para receber pacotes. Teremos mais a dizer sobre eles quando descrevemos o TCP, que também usa portas. Na verdade, o valor principal de UDP em vez de apenas usar IP bruto é a adição das portas de origem e de destino. Sem os campos da porta, a camada de transporte não saberia o que fazer com cada pacote de entrada. Com eles, ele entrega o segmento incorporado ao aplicativo correto plicatura.

32 bits
Porta de origem
Comprimento UDP
Porto de destino
UDP checksum

Figura 6-27. O cabeçalho UDP.

A porta de origem é principalmente necessária quando uma resposta deve ser enviada de volta ao fonte. Ao copiar o campo *Porta de origem* do segmento de entrada para o Campo da *porta de destino* do segmento de saída, o processo de envio da resposta pode especificar qual processo na máquina de envio deve obtê-lo. O campo de *comprimento UDP* inclui o cabeçalho de 8 bytes e os dados. O mínimo comprimento é de 8 bytes, para cobrir o cabeçalho. O comprimento máximo é de 65.515 bytes, que é menor do que o maior número que caberá em 16 bits por causa do limite de tamanho em Pacotes IP.

Um *checksum* opcional também é fornecido para maior confiabilidade. Ele faz a soma de verificação cabeçalho, os dados e um pseudo-cabeçalho de IP conceitual. Ao realizar este com , o campo *Checksum* é definido como zero e o campo de dados é preenchido com um byte zero adicional se seu comprimento for um número ímpar. O algoritmo de checksum é simplesmente somar todas as palavras de 16 bits em um complemento e tomar complemento da soma. Como consequência, quando o receptor realiza o cálculo em todo o segmento, incluindo o campo *Checksum*, o resultado deve ser 0.

Se a soma de verificação não for calculada, ela é armazenada como 0, pois por uma feliz coincidência da aritmética do complemento de alguém, um 0 calculado verdadeiro é armazenado como todos os 1s. Contudo, desligá-lo é tolice, a menos que a qualidade dos dados não importe (por exemplo, para digí-
discurso padronizado). O pseudoheader para o caso do IPv4 é mostrado na Figura 6.28. Contém o Endereços IPv4 de 32 bits das máquinas de origem e destino, o número do protocolo para UDP (17), e a contagem de bytes para o segmento UDP (incluindo o cabeçalho). isto

Página 567

SEC. 6,4
OS PROTOCOLOS DE TRANSPORTE DA INTERNET: UDP

543

é diferente, mas análogo para IPv6. Incluindo o pseudoheader no UDP cálculo de checksum ajuda a detectar pacotes mal entregues, mas incluindo também viola a hierarquia de protocolo, uma vez que os endereços IP nele pertencem à camada IP, não para a camada UDP. O TCP usa o mesmo pseudoheader para sua soma de verificação.

32 bits
Endereço de Origem
Endereço de destino
0 0 0 0 0 0 0
Protocolo = 17
Comprimento UDP

Figura 6-28. O pseudoheader IPv4 incluído na soma de verificação UDP.

Provavelmente, vale a pena mencionar explicitamente algumas das coisas que o UDP faz *não* faça. Não faz controle de fluxo, controle de congestionamento ou retransmissão mediante recebimento de um segmento inválido. Tudo isso depende dos processos do usuário. O que isso faz é fornecer uma interface para o protocolo IP com o recurso adicional de demultiplexação vários processos usando as portas e detecção de erro ponta a ponta opcional. Isso é tudo o que faz.

Para aplicativos que precisam ter controle preciso sobre o fluxo de pacotes, erro controle, ou tempo, UDP fornece apenas o que o médico receitou. Uma área onde é especialmente útil em situações cliente-servidor. Muitas vezes, o cliente envia uma breve resposta indaga ao servidor e espera uma resposta curta. Se o pedido ou o a resposta for perdida, o cliente pode simplesmente encerrar o tempo e tentar novamente. Não é apenas o código sim-

ple, mas menos mensagens são necessárias (uma em cada direção) do que com um protocolo exigindo uma configuração inicial como TCP.

Um aplicativo que usa UDP dessa forma é o DNS (Domain Name System), que estudaremos no cap. 7. Em resumo, um programa que precisa procurar o IP endereço de algum nome de host, por exemplo, www.cs.berkeley.edu, pode enviar um UDP pacote contendo o nome do host para um servidor DNS. O servidor responde com um UDP pacote contendo o endereço IP do host. Nenhuma configuração é necessária com antecedência e não o aluguel é necessário posteriormente. Apenas duas mensagens passam pela rede.

6.4.2 Chamada de procedimento remoto

Em certo sentido, enviar uma mensagem a um host remoto e obter uma resposta é muito parecido com fazer uma chamada de função em uma linguagem de programação. Em ambos os casos, você

comece com um ou mais parâmetros e você obterá um resultado. Esta observação tem levou as pessoas a tentar organizar interações de solicitação-resposta nas redes para serem transmitidas no

Página 568

544
A CAMADA DE TRANSPORTE
INDIVÍDUO. 6

forma de chamadas de procedimento. Tal arranjo torna os aplicativos de rede muito

mais fácil de programar e mais familiar de lidar. Por exemplo, imagine um procedimento denominado *obter endereço IP (nome do host)* que funciona enviando um UDP pacote para um servidor DNS e aguardando a resposta, atingindo o tempo limite e tentando novamente se um não chega rápido o suficiente. Desta forma, todos os detalhes da rede pode ser ocultado do programador.

O trabalho chave nesta área foi feito por Birrell e Nelson (1984). Em uma noz-shell, o que Birrell e Nelson sugeriram foi permitir que os programas chamem processos localizados em hosts remotos. Quando um processo na máquina 1 chama um procedimento em máquina 2, o processo de chamada em 1 é suspenso e a execução do chamado pro a transferência ocorre em 2. As informações podem ser transportadas do chamador para a chamada lee nos parâmetros e pode voltar no resultado do procedimento. Sem mensagem pas-sing fica visível para o programador do aplicativo. Esta técnica é conhecida como **RPC** (**Chamada de procedimento remoto**) e se tornou a base para muitos aplicativos de rede cátions. Tradicionalmente, o procedimento de chamada é conhecido como cliente e o chamado procedimento é conhecido como servidor e usaremos esses nomes aqui também.

A ideia por trás do RPC é fazer uma chamada de procedimento remoto parecer tanto quanto possível como um local. Na forma mais simples, para chamar um procedimento remoto, o cliente programa deve ser vinculado a um pequeno procedimento de biblioteca, chamado **stub do cliente**, que representa o procedimento do servidor no espaço de endereço do cliente. Da mesma forma, o servidor está vinculado a um procedimento denominado **stub do servidor**. Esses procedimentos escondem o fato que a chamada de procedimento do cliente para o servidor não é local.

As etapas reais para fazer um RPC são mostradas na Figura 6.29. O passo 1 é o cli-ent chamando o stub do cliente. Esta chamada é uma chamada de procedimento local, com os parâmetros empurrado para a pilha da maneira normal. A etapa 2 é o esboço do cliente embalando o pa-rameters em uma mensagem e fazendo uma chamada de sistema para enviar a mensagem. Embalagem os parâmetros são chamados de **empacotamento**. A etapa 3 é o sistema operacional enviando o mensagge da máquina cliente para a máquina servidor. A etapa 4 é a operação sistema passando o pacote de entrada para o stub do servidor. Finalmente, a etapa 5 é o servidor stub chamando o procedimento do servidor com os parâmetros não empacotados. A resposta traça o mesmo caminho na outra direção.

O principal item a ser observado aqui é que o procedimento do cliente, escrito pelo usuário, apenas faz uma chamada de procedimento normal (ou seja, local) para o stub do cliente, que tem o mesmo nome como o procedimento do servidor. Uma vez que o procedimento do cliente e o stub do cliente estão no mesmo espaço de endereço, os parâmetros são passados da maneira usual. Da mesma forma, o o procedimento do servidor é chamado por um procedimento em seu espaço de endereço com os parâmetros ele espera. Para o procedimento do servidor, nada é incomum. Desta forma, em vez de I / O sendo feito em soquetes, a comunicação de rede é feita falsificando um normal chamada de procedimento.

Apesar da elegância conceitual do RPC, existem algumas cobras escondidas sob a grama. Um grande problema é o uso de parâmetros de ponteiro. Normalmente, passar um ponteiro para um procedimento não é um problema. O procedimento chamado pode usar o ponteiro no da mesma forma que o chamador pode porque ambos os procedimentos vivem no mesmo endereço virtual

```

CPU cliente
Cliente
toco
Cliente
2
1
Sistema operacional
CPU do servidor
Servidor
toco
4
3
5
Sistema operacional
Servidor
Rede

```

Figura 6-29. Etapas para fazer uma chamada de procedimento remoto. Os tocos estão sombreados. espaço. Com RPC, passar ponteiros é impossível porque o cliente e o servidor são em diferentes espaços de endereço.

Em alguns casos, truques podem ser usados para tornar possível passar ponteiros. Supõe-se que o primeiro parâmetro é um ponteiro para um inteiro, k . O stub do cliente pode marcar k e envie-o para o servidor. O stub do servidor cria um ponteiro para k que passa para o procedimento do servidor, exatamente como ele espera. Quando o servidor procede, dure retorna o controle para o stub do servidor, o último envia k de volta para o cliente, onde o novo k é copiado sobre o antigo, caso o servidor o altere. Com efeito, a sequência de chamada padrão de chamada por referência foi substituída por chamada por cópia-restauração. Infelizmente, esse truque nem sempre funciona, por exemplo, se o ponteiro aponta para um gráfico ou outra estrutura de dados complexa. Por este motivo, algumas restrições devem ser colocadas em parâmetros para procedimentos chamados remotamente, como nós deve ver.

Um segundo problema é que em linguagens de digitação fraca, como C, é perfeitamente legal escrever um procedimento que calcula o produto interno de dois vetores (matrizes), sem especificar o tamanho de cada um. Cada um poderia ser encerrado por um especial valor conhecido apenas pelos procedimentos de chamada e chamados. Sob essas circunstâncias posições, é essencialmente impossível para o stub do cliente empacotar os parâmetros: não tem como determinar o tamanho deles.

Um terceiro problema é que nem sempre é possível deduzir os tipos de parâmetros, nem mesmo de uma especificação formal ou do próprio código. Um exemplo é *printf*, que pode ter qualquer número de parâmetros (pelo menos um), e o parâmetro *ters* podem ser uma mistura arbitrária de inteiros, shorts, longos, caracteres, strings, float-números de pontos de vários comprimentos e outros tipos. Tentando chamar *printf* como um o procedimento remoto seria praticamente impossível porque C é muito permissivo. No entanto, uma regra dizendo que RPC pode ser usado, desde que você não programe C (ou C++) não seria popular com muitos programadores.

546

A CAMADA DE TRANSPORTE INDIVÍDUO. 6

Um quarto problema está relacionado ao uso de variáveis globais. Normalmente, a chamada e o procedimento chamado pode se comunicar usando variáveis globais, além de comunicando por meio de parâmetros. Mas se o procedimento chamado for movido para um controle remoto

máquina, o código falhará porque as variáveis globais não são mais compartilhadas.

Esses problemas não pretendem sugerir que o RPC seja impossível. Na verdade, é amplamente utilizado, mas algumas restrições são necessárias para que funcione bem na prática. Em termos de protocolos da camada de transporte, o UDP é uma boa base para a implementação de RPC. Tanto as solicitações quanto as respostas podem ser enviadas como um único pacote UDP no

caso mais simples e a operação pode ser rápida. No entanto, uma implementação deve incluir outras máquinas também. Porque o pedido ou a resposta podem ser perdidos, o

o cliente deve manter um cronômetro para retransmitir a solicitação. Observe que uma resposta serve como um

confirmação implícita de uma solicitação, portanto, a solicitação não precisa ser separada reconhecido. Às vezes, os parâmetros ou resultados podem ser maiores do que o máximo tamanho do pacote UDP, caso em que algum protocolo é necessário para entregar grandes mensagens. Se várias solicitações e respostas podem se sobrepor (como no caso de concorrer programação de aluguel), um identificador é necessário para combinar a solicitação com a resposta. Uma preocupação de alto nível é que a operação pode não ser idempotente (ou seja, segura repetir). O caso simples são operações idempotentes, como solicitações de DNS e respostas. O cliente pode retransmitir com segurança essas solicitações repetidas vezes se não as respostas estão disponíveis. Não importa se o servidor nunca recebeu o pedido, ou foi a resposta que foi perdida. A resposta, quando finalmente chegar, ser o mesmo (supondo que o banco de dados DNS não seja atualizado nesse interim). Como nunca, nem todas as operações são idempotentes, por exemplo, porque têm importantes efeitos colaterais, como incrementar um contador. RPC para essas operações requer semântica mais forte de modo que quando o programador chama um procedimento ele não é executado

cortado várias vezes. Neste caso, pode ser necessário configurar uma conexão TCP e enviar a solicitação sobre ele em vez de usar UDP.

6.4.3 Protocolos de Transporte em Tempo Real

RPC cliente-servidor é uma área em que o UDP é amplamente utilizado. Outro é para aplicativos de multimídia em tempo real. Em particular, como rádio na Internet, Internet telefonia, música sob demanda, videoconferência, vídeo sob demanda e outros aplicativos timedia se tornaram mais comuns, as pessoas descobriram que cada aplicação estava reinventando mais ou menos o mesmo protocolo de transporte em tempo real col. Gradualmente ficou claro que ter um protocolo de transporte em tempo real genérico para vários aplicativos seria uma boa ideia.

Assim nasceu o **RTP (Real-time Transport Protocol)**. É descrito em RFC 3550 e agora é amplamente utilizado para aplicações multimídia. Vamos descrever dois aspectos do transporte em tempo real. O primeiro é o protocolo RTP para transporte dados de áudio e vídeo em pacotes. O segundo é o processamento que ocorre, principalmente no receptor, para reproduzir o áudio e o vídeo no momento certo. Estes as funções se encaixam na pilha de protocolo, conforme mostrado na Figura 6-30.

Página 571

SEC. 6,4

OS PROTOCOLOS DE TRANSPORTE DA INTERNET: UDP

547

Aplicativo multimídia

RTP

Interface de soquete

UDP

IP

Ethernet

(uma)

(b)

Ethernet

cabeçalho

IP

cabeçalho

UDP

cabeçalho

RTP

cabeçalho

RTP payload

Carga útil UDP

Carga de IP

Carga útil Ethernet

Do utilizador

espaço

SO

Núcleo

Figura 6-30. (a) A posição do RTP na pilha do protocolo. (b) Aninhamento de pacotes.

O RTP normalmente é executado no espaço do usuário sobre UDP (no sistema operacional). Funciona

ates como segue. O aplicativo de multimídia consiste em vários áudio, vídeo, texto e, possivelmente, outros fluxos. Estes são alimentados na biblioteca RTP, que está em espaço do usuário junto com o aplicativo. Esta biblioteca multiplexa os fluxos e codifica-os em pacotes RTP, que são colocados em um soquete. No sistema operacional lado do soquete, os pacotes UDP são gerados para envolver os pacotes RTP e entregue ao IP para transmissão por um link como Ethernet. O processo reverso acontece no receptor. O aplicativo de multimídia eventualmente recebe vários dados de mídia da biblioteca RTP. É responsável por representar a mídia. A pilha de protocolos para essa situação é mostrada na Figura 6.30 (a). O aninhamento de pacotes é mostrado na Fig. 6-30 (b).

Como consequência deste design, é um pouco difícil dizer qual camada RTP é. In. Uma vez que é executado no espaço do usuário e está vinculado ao programa de aplicação, certamente

parece um protocolo de aplicativo. Por outro lado, é um aplicativo genérico protocolo independente que apenas fornece facilidades de transporte, por isso também se parece com um

protocolo de transporte. Provavelmente, a melhor descrição é que é um protocolo de transporte que apenas acontece de ser implementado na camada de aplicativo, é por isso que estamos cobrindo-o neste capítulo.

RTP - O protocolo de transporte em tempo real

A função básica do RTP é multiplexar vários fluxos de dados em tempo real em um único fluxo de pacotes UDP. O fluxo UDP pode ser enviado para um único destino (unicast) ou para vários destinos (multicast). Porque RTP apenas usa UDP normal, seus pacotes não são tratados de maneira especial pelos roteadores, a menos que alguns

recursos de qualidade de serviço de IP incorretos estão ativados. Em particular, não há garantias sobre a entrega, e os pacotes podem ser perdidos, atrasados, corrompidos, etc.

O formato RTP contém vários recursos para ajudar os receptores a trabalhar com informações sobre a mídia. Cada pacote enviado em um fluxo RTP recebe um número um

Página 572

548

A CAMADA DE TRANSPORTE INDIVÍDUO. 6

maior do que seu antecessor. Esta numeração permite que o destino determine se quaisquer pacotes estão faltando. Se um pacote estiver faltando, a melhor ação para o destino levar depende da aplicação. Pode ser para pular um quadro de vídeo se os pacotes forem transportando dados de vídeo, ou para aproximar o valor ausente por interpolação se os pacotes transportam dados de áudio. A retransmissão não é uma opção prática, uma vez que o pacote retransmitido provavelmente chegaria tarde demais para ser útil. Como consequência sequência, RTP não tem confirmações e nenhum mecanismo para solicitar retransmissão sessões.

Cada carga útil RTP pode conter várias amostras, e elas podem ser codificadas em qualquer forma que o aplicativo deseja. Para permitir a interoperação, RTP define vários perfis (por exemplo, um único fluxo de áudio), e para cada perfil, codificação múltipla para-tapetes podem ser permitidos. Por exemplo, um único fluxo de áudio pode ser codificado como 8-bit amostras PCM a 8 kHz usando codificação delta, codificação preditiva, en-codificação, codificação de MP3 e assim por diante. RTP fornece um campo de cabeçalho no qual a fonte pode especificar a codificação, mas de outra forma não está envolvida em como a codificação é feito.

Outro recurso de que muitos aplicativos em tempo real precisam é o carimbo de data / hora. A ideia aqui é permitir que a fonte associe um timestamp com a primeira amostra em cada pacote. Os carimbos de data / hora são relativos ao início do stream, então apenas a diferença as diferenças entre os carimbos de data / hora são significativas. Os valores absolutos não têm média ing. Como descreveremos em breve, este mecanismo permite que o destino faça um pequena quantidade de buffer e reproduz cada amostra no número certo de milissegundos

após o início do stream, independentemente de quando o pacote contendo o sample chegou.

Não só o timestamping reduz os efeitos da variação no atraso da rede, mas também permite que vários fluxos sejam sincronizados entre si. Para exemplo, um programa de televisão digital pode ter um stream de vídeo e dois de áudio cárregos. Os dois fluxos de áudio podem ser para transmissões estéreo ou para lidar com filmes com trilha sonora no idioma original e trilha sonora dobrada para o local idioma, dando ao espectador uma escolha. Cada fluxo vem de um físico diferente dispositivo de chamada, mas se eles forem marcados com data e hora de um único contador, eles podem ser reproduzidos de volta sincronadamente, mesmo se os fluxos forem transmitidos e / ou recebidos de alguma forma erraticamente.

O cabeçalho RTP é ilustrado na Figura 6-31. Consiste em três palavras de 32 bits e potencialmente algumas extensões. A primeira palavra contém o campo *Versão*, que já está em 2. Esperemos que esta versão esteja muito próxima da versão final, pois existe apenas um ponto de código restante (embora 3 possa ser definido como significando que o versão real estava em uma palavra de extensão).

O bit *P* indica que o pacote foi preenchido com um múltiplo de 4 bytes.

O último byte de preenchimento informa quantos bytes foram adicionados. O bit *X* indica que um cabeçalho de extensão está presente. O formato e o significado do cabeçalho da extensão não estão definidos. A única coisa que é definida é que a primeira palavra da extensão dá o comprimento. Esta é uma saída de emergência para quaisquer requisitos imprevistos.

Página 573

SEC. 6,4

OS PROTOCOLOS DE TRANSPORTE DA INTERNET: UDP

549

32 bits

Ver. PX

M

Tipo de carga útil

Número sequencial

Timestamp

Identificador de fonte de sincronização

Contribuindo com identificador de fonte

CC

Figura 6-31. O cabeçalho RTP.

O campo *CC* informa quantas fontes contribuintes estão presentes, de 0 a 15 (ver abaixo). O bit *M* é um bit marcador específico do aplicativo. Pode ser usado para marcar o início de um quadro de vídeo, o início de uma palavra em um canal de áudio, ou algo outra coisa que o aplicativo entende. O campo *Payload type* informa qual algoritmo de codificação foi usado (por exemplo, áudio descompactado de 8 bits, MP3, etc.). Como todo pacote carrega esse campo, a codificação pode mudar durante a transmissão.

O *número de sequência* é apenas um contador que é incrementado em cada pacote RTP enviei. É usado para detectar pacotes perdidos.

O *carimbo de data / hora* é produzido pela fonte do fluxo para observar quando o primeiro sample no pacote foi feito. Este valor pode ajudar a reduzir a variabilidade de tempo chamada jitter no receptor, desacoplando a reprodução do tempo de chegada do pacote. O

O identificador da fonte de sincronização informa a qual fluxo o pacote pertence. Isto é o método usado para multiplexar e demultiplexar múltiplos fluxos de dados em um único fluxo de pacotes UDP. Finalmente, os *identificadores de fonte contribuintes*, se houver, são usado quando mixers estão presentes no estúdio. Nesse caso, o mixer é o syncronizada e os fluxos sendo misturados estão listados aqui.

RTCP - O protocolo de controle de transporte em tempo real

O RTP tem um protocolo de irmã mais nova (protocolo de irmã menor?) Chamado **RTCP (Real-Protocolo de Controle de Transporte de tempo)**. É definido junto com RTP em RFC 3550 e lida com feedback, sincronização e interface do usuário. Não transmite qualquer amostra de mídia.

A primeira função pode ser usada para fornecer feedback sobre o atraso, variação no atraso

ou jitter, largura de banda, congestionamento e outras propriedades de rede para as fontes. Isto as informações podem ser usadas pelo processo de codificação para aumentar a taxa de dados (e dar melhor qualidade) quando a rede está funcionando bem e para cortar os dados

550

A CAMADA DE TRANSPORTE

INDIVÍDUO. 6

taxa quando há problemas na rede. Ao fornecer feedback contínuo, os algoritmos de codificação podem ser continuamente adaptados para fornecer a melhor qualidade pós-sível nas atuais circunstâncias. Por exemplo, se a largura de banda aumentar ou diminuir durante a transmissão, a codificação pode mudar de MP3 para 8 bits PCM para codificação delta conforme necessário. O campo *Payload type* é usado para informar o destino que qual algoritmo de codificação é usado para o pacote atual, tornando possível variar sob demanda.

Um problema com o fornecimento de feedback é que os relatórios RTCP são enviados para todos os participantes. Para uma aplicação multicast com um grande grupo, a largura de banda usada por O RTCP aumentaria rapidamente. Para evitar que isso aconteça, os remetentes RTCP reduzem a taxa de seus relatórios para consumir coletivamente não mais do que, digamos, 5% da largura de banda da mídia. Para fazer isso, cada participante precisa conhecer a mídia largura de banda, que ele aprende com o remetente, e o número de participantes, que estima ouvindo outros relatórios RTCP.

O RTCP também lida com a sincronização entre fluxos. O problema é tão diferente streams podem usar relógios diferentes, com granularidades diferentes e derivar diferentes cotações. O RTCP pode ser usado para mantê-los sincronizados.

Finalmente, o RTCP fornece uma maneira de nomear as várias fontes (por exemplo, em ASCII texto). Esta informação pode ser exibida na tela do receptor para indicar quem está falando no momento.

Mais informações sobre RTP podem ser encontradas em Perkins (2003).

Playout com Buffer e Controle de Jitter

Uma vez que as informações da mídia chegam ao receptor, devem ser reproduzidas no tempo certo. Em geral, este não será o momento em que o pacote RTP chegou o receptor porque os pacotes levarão quantidades ligeiramente diferentes de tempo para transitar a rede. Mesmo que os pacotes sejam injetados exatamente com os intervalos corretos, entre eles no remetente, eles chegarão ao receptor com diferentes pausas. Essa variação no atraso é chamada de **jitter**. Mesmo uma pequena quantidade de jitter de pacote

pode causar artefatos de mídia que distraem, como quadros de vídeo irregulares e ininteligíveis áudio, se a mídia for simplesmente reproduzida à medida que chega.

A solução para este problema é **armazenar os pacotes** no receptor antes que eles são executados para reduzir o jitter. Como exemplo, na Figura 6-32, vemos um fluxo de pacotes sendo entregues com uma quantidade substancial de jitter. O pacote 1 é enviado de o servidor em $t = 0$ seg e chega ao cliente em $t = 1$ seg. O pacote 2 sofre mais demora e leva 2 segundos para chegar. Conforme os pacotes chegam, eles são armazenados em buffer

a máquina cliente.

Em $t = 10$ seg, a reprodução começa. Neste momento, os pacotes 1 a 6 foram armazenados em buffer para que possam ser removidos do buffer em intervalos uniformes durante jogo suave. No caso geral, não é necessário usar intervalos uniformes antes de porque os carimbos de data / hora RTP informam quando a mídia deve ser reproduzida.

551

```
0  
5  
1 2 3 4 5 6 7  
8  
10  
Tempo (seg)  
Tempo no buffer  
15  
20  
Intervalo na reprodução  
1  
Pacote removido do buffer  
Pacote chega ao buffer  
2  
3 4 5  
6  
7  
8  
1 2 3 4 5 6 7 8  
Packetdepartsource
```

Figura 6-32. Suavizando o fluxo de saída armazenando pacotes em buffer.

Infelizmente, podemos ver que o pacote 8 está tão atrasado que está não está disponível quando o slot de jogo aparece. Existem duas opções. O pacote 8 pode ser pulado e o jogador pode passar para os pacotes subsequentes. Alternativamente, jogue volta pode parar até que o pacote 8 chegue, criando uma lacuna irritante na música ou filme. Em um aplicativo de mídia ao vivo, como uma chamada de voz sobre IP, o pacote será ignorado. Aplicativos Live não funcionam bem em espera. Em um streaming media application dia, o jogador pode pausar. Este problema pode ser aliviado por atrasos o tempo de início ainda mais, usando um buffer maior. Para um streaming de áudio ou player de vídeo, buffers de cerca de 10 segundos são freqüentemente usados para garantir que o player recebe todos os pacotes (que não são descartados na rede) a tempo. Para viver aplicativos como videoconferência, buffers curtos são necessários para a capacidade de resposta. Uma consideração importante para uma reprodução suave é o **ponto de reprodução**, ou por quanto tempo aguarde a mídia no receptor antes de reproduzi-la. Decidindo quanto tempo esperar depende do jitter. A diferença entre um jitter baixo e alto jitter com conexão é mostrada na Fig. 6-33. O atraso médio pode não diferir muito entre os dois, mas se houver alta instabilidade, o ponto de reprodução pode precisar ser muito mais longe para capturar 99% dos pacotes do que se houver baixo jitter.

Para escolher um bom ponto de reprodução, o aplicativo pode medir o jitter olhando na diferença entre os timestamps RTP e a hora de chegada. Cada diferença fornece uma amostra do atraso (mais um deslocamento fixo arbitrário). No entanto, o atraso pode mudar com o tempo devido a outros tráfegos concorrentes e rotas em mudança. Para acomodar essa mudança, os aplicativos podem adaptar seu ponto de reprodução enquanto eles estão correndo. No entanto, se não for bem feito, alterar o ponto de reprodução pode trazer uma falha observável para o usuário. Uma maneira de evitar este problema para áudio é para adaptar o ponto de reprodução entre os **surtos de fala**, nas lacunas de uma conversa. Não notará a diferença entre um silêncio curto e um silêncio um pouco mais longo. RTP permite que os aplicativos definam o bit marcador *M* para indicar o início de um novo surto de fala para este propósito.

Se o atraso absoluto até a reprodução da mídia for muito longo, os aplicativos ativos sofrerá. Nada pode ser feito para reduzir o atraso de propagação se um caminho direto for

552

A CAMADA DE TRANSPORTE

INDIVÍDUO. 6

Alta instabilidade

Baixo jitter

Mínimo

demora

(devido à velocidade da luz)

Demora

(uma)
F
ação
do
pac
k
ets
F
ação
do
pac
k
ets
Demora
(b)

Figura 6-33. (a) Alta instabilidade. (b) Jitter baixo.

já está sendo usado. O ponto de reprodução pode ser puxado simplesmente aceitando que uma fração maior de pacotes chegará tarde demais para serem reproduzidos. Se isso não for aceitável, a única maneira de puxar o ponto de reprodução é reduzir o jitter usando uma melhor qualidade de serviço, por exemplo, o encaminhamento rápido diferenciado serviço. Ou seja, é necessária uma rede melhor.

6.5 OS PROTOCOLOS DE TRANSPORTE DA INTERNET: TCP

UDP é um protocolo simples e tem alguns usos muito importantes, como cliente interações de servidor e multimídia, mas para a maioria das aplicações da Internet, confiável, se entrega interrompida é necessária. UDP não pode fornecer isso, então outro protocolo é exigido. É chamado de TCP e é o principal carro-chefe da Internet. Deixe-nos agora estude-o em detalhes.

6.5.1 Introdução ao TCP

TCP (Transmission Control Protocol) foi projetado especificamente para fornecer um fluxo de bytes de ponta a ponta confiável em uma internetwork não confiável. Uma internet o trabalho difere de uma única rede porque diferentes partes podem ter grandes diferenças diferentes topologias, larguras de banda, atrasos, tamanhos de pacote e outros parâmetros. TCP foi projetado para se adaptar dinamicamente às propriedades da internetwork e ser robusto em face de muitos tipos de falhas.

O TCP foi definido formalmente na RFC 793 em setembro de 1981. Com o passar do tempo, muitas melhorias foram feitas e vários erros e inconsistências foram corrigido. Para lhe dar uma ideia da extensão do TCP, as RFCs importantes são

SEC. 6,5
OS PROTOCOLOS DE TRANSPORTE DA INTERNET: TCP

553

agora RFC 793 plus: esclarecimentos e correções de bugs no RFC 1122; extensões para alto desempenho em RFC 1323; reconhecimentos seletivos na RFC 2018; vigarista-controle de gestação no RFC 2581; reaproveitamento de campos de cabeçalho para qualidade de serviço em

RFC 2873; temporizadores de retransmissão aprimorados no RFC 2988; e congestionamento explícito

notificação no RFC 3168. A coleção completa é ainda maior, o que levou a um guia às muitas RFCs, publicadas, é claro, como outro documento RFC, RFC 4614.

Cada máquina que suporta TCP tem uma entidade de transporte TCP, seja uma biblioteca cedura, um processo do usuário ou, mais comumente, parte do kernel. Em todos os casos, ele envelhece fluxos e interfaces TCP para a camada IP. Uma entidade TCP aceita dados do usuário fluxos de processos locais, divide-os em partes não superiores a 64 KB (em prática, muitas vezes 1460 bytes de dados para caber em um único quadro Ethernet com o Cabeçalhos IP e TCP) e envia cada parte como um datagrama IP separado. Quando datagramas contendo dados TCP chegam a uma máquina, eles são dados ao TCP entity, que reconstrói os fluxos de bytes originais. Para simplificar, vamos algumas vezes usam apenas "TCP" para significar a entidade de transporte TCP (um pedaço de software) ou o protocolo TCP (um conjunto de regras). A partir do contexto, ficará claro o que isso significa. Por exemplo, em "O usuário fornece os dados ao TCP", a entidade de transporte TCP é clara-pretendia.

A camada IP não dá nenhuma garantia de que os datagramas serão entregues corretamente, nem qualquer indicação de quanto rápido os datagramas podem ser enviados. Cabe ao TCP enviar dados rapidamente o suficiente para fazer uso da capacidade, mas não causar congestionamento, e para expirar e retransmitir quaisquer datagramas que não sejam entregues. Datagramas que fazem chegar pode muito bem fazê-lo na ordem errada; também cabe ao TCP remontá-los em mensagens na sequência adequada. Em suma, o TCP deve fornecer bom desempenho com a confiabilidade que a maioria dos aplicativos deseja e que o IP não oferece vide.

6.5.2 O Modelo de Serviço TCP

O serviço TCP é obtido tanto pelo remetente quanto pelo receptor, criando o fim pontos, chamados **soquetes**, conforme discutido na Seç. 6.1.3. Cada soquete tem um número de soquete

ber (endereço) que consiste no endereço IP do host e um número de 16 bits local para esse host, chamado de **porta**. Uma porta é o nome TCP de um TSAP. Para serviço TCP para ser obtida, uma conexão deve ser explicitamente estabelecida entre um soquete em um

máquina e um soquete em outra máquina. As chamadas de soquete estão listadas na Figura 6-5.

Um soquete pode ser usado para várias conexões ao mesmo tempo. Em outras palavras, duas ou mais conexões podem terminar no mesmo soquete. Conexões

são identificados pelos identificadores de soquete em ambas as extremidades, ou seja,

(*soquete1*, *soquete2*). Não

números de circuitos virtuais ou outros identificadores são usados.

Números de porta abaixo de 1024 são reservados para serviços padrão que geralmente podem ser iniciados apenas por usuários com privilégios (por exemplo, root em sistemas UNIX). Eles são chamados

portos bem conhecidos. Por exemplo, qualquer processo que deseja recuperar remotamente o correio

de um host pode se conectar à porta 143 do host de destino para entrar em contato com seu IMAP

Página 578

554

A CAMADA DE TRANSPORTE

INDIVÍDUO. 6

daemon. A lista de portos conhecidos é fornecida em www.iana.org. Mais de 700 têm sido atribuídos. Alguns dos mais conhecidos estão listados na Figura 6-34.

Porta

Protocolo

Usar

20, 21

FTP

Transferência de arquivo

22

SSH

Login remoto, substituição de Telnet

25

SMTP

O email

80

HTTP

Rede mundial de computadores

110

POP-3

Acesso remoto de e-mail

143

IMAP

Acesso remoto de e-mail

443

HTTPS

Web segura (HTTP sobre SSL / TLS)

543

RTSP

Controle do media player

631

IPP

Compartilhamento de impressora

Figura 6-34. Algumas portas atribuídas.

Outras portas de 1024 a 49151 podem ser registradas com IANA para uso por usuários sem privilégios, mas os aplicativos podem escolher e escolhem suas próprias portas. Para ex-

amplamente, o aplicativo de compartilhamento de arquivos ponto a ponto BitTorrent (não oficialmente) usa

portas 6881–6887, mas pode ser executado em outras portas também.

Certamente seria possível ter o daemon FTP anexado à porta 21

na hora da inicialização, o daemon SSH se conecta à porta 22 na hora da inicialização e assim por diante.

No entanto, fazer isso iria sobrecarregar a memória com daemons que estavam ociosos na maioria das

A Hora. Em vez disso, o que normalmente é feito é ter um único daemon, chamado **inetd** (**I**nternet **d**aemon) no UNIX, conecta-se a várias portas e aguarde o primeira conexão de entrada. Quando isso ocorre, o *inetd* desvia um novo processo e executa o daemon apropriado nele, permitindo que esse daemon cuide da solicitação. No desta forma, os daemons diferentes do *inetd* só estão ativos quando há trabalho para eles fazem. O Inetd aprende quais portas ele deve usar a partir de um arquivo de configuração. Conse-

frequentemente, o administrador do sistema pode configurar o sistema para ter dados permanentes mons nas portas mais ocupadas (por exemplo, porta 80) e *inetd* nas demais.

Todas as conexões TCP são full duplex e ponto a ponto. Full duplex significa que o tráfego pode ir nas duas direções ao mesmo tempo. Ponto a ponto significa que cada conexão tem exatamente dois pontos finais. TCP não suporta multicast ou transmissão.

Uma conexão TCP é um fluxo de bytes, não um fluxo de mensagens. Mensagem vinculada áries não são preservadas de ponta a ponta. Por exemplo, se o processo de envio faz quatro Gravações de 512 bytes em um fluxo TCP, esses dados podem ser entregues ao receptor processar como quatro pedaços de 512 bytes, dois pedaços de 1024 bytes, um pedaço de 2048 bytes (ver

Fig. 6-35), ou de alguma outra forma. Não há como o receptor detectar o unidade (s) em que os dados foram gravados, não importa o quanto tente.

Página 579

SEC. 6,5

OS PROTOCOLOS DE TRANSPORTE DA INTERNET: TCP

55

UMA

B

C

D

ABCD

Cabeçalho IP

Cabeçalho TCP

(uma)

(b)

Figura 6-35. (a) Quatro segmentos de 512 bytes enviados como datagramas IP separados. (b) O 2048 bytes de dados entregues ao aplicativo em uma única chamada READ .

Os arquivos no UNIX também têm essa propriedade. O leitor de um arquivo não sabe se o arquivo foi escrito um bloco por vez, um byte por vez ou tudo de uma vez. Como com um arquivo UNIX, o software TCP não tem ideia do que os bytes significam e não tem interesse em descobrir. Um byte é apenas um byte.

Quando um aplicativo passa dados para o TCP, o TCP pode enviá-los imediatamente ou armazená-lo em buffer (para coletar uma quantia maior para enviar de uma vez), a seu critério.

No entanto, às vezes o aplicativo realmente deseja que os dados sejam enviados imediatamente.

Por exemplo, suponha que um usuário de um jogo interativo deseja enviar um fluxo de atualizações. É essencial que as atualizações sejam enviadas imediatamente, não armazenadas em buffer até

há uma coleção deles. Para forçar a saída de dados, o TCP tem a noção de um PUSH

sinalizador que é transportado em pacotes. A intenção original era permitir que os aplicativos avisassem ao TCP implementações através do flag PUSH para não atrasar a transmissão. No entanto, aplicações não podem definir literalmente o sinalizador PUSH quando enviam dados. Em vez disso, diferentes sistemas operacionais desenvolveram diferentes opções para agilizar a transmissão (por exemplo, TCP NODELAY no Windows e Linux). Para os arqueólogos da Internet, também mencionaremos uma característica interessante do Serviço TCP que permanece no protocolo, mas raramente é usado: **dados urgentes**. Quando um aplicativo tem dados de alta prioridade que devem ser processados imediatamente, por exemplo, se um usuário interativo pressionar a tecla CTRL-C para interromper um computador remoto no fluxo de dados e fornecê-lo ao TCP junto com o sinalizador URGENTE. Isto evento faz com que o TCP pare de acumular dados e transmita tudo o que tem para isso conexão imediatamente. Quando os dados urgentes são recebidos no destino, o aplicativo receptor é interrompido (por exemplo, dado um sinal em termos do UNIX) para que possa parar o que quer que seja fazer e ler o fluxo de dados para encontrar os dados urgentes. O fim dos dados urgentes é marcado para que o aplicativo saiba quando terminar. O início dos dados urgentes é não marcado. Cabe ao aplicativo descobrir isso. Este esquema fornece um mecanismo de sinalização bruto e deixa todo o resto até o aplicativo. No entanto, embora os dados urgentes sejam potencialmente úteis, não encontrou aplicação convincente no início e caiu em desuso. Seu uso agora é desencorajado por causa das diferenças de implementação, deixando os aplicativos para lidar com seus próprios sinalização. Talvez os protocolos de transporte futuros forneçam melhor sinalização.

Página 580

556

A CAMADA DE TRANSPORTE
INDIVÍDUO. 6

6.5.3 O Protocolo TCP

Nesta seção, daremos uma visão geral do protocolo TCP. No a seguir, examinaremos o cabeçalho do protocolo, campo por campo. Uma característica fundamental do TCP, e que domina o design do protocolo, é que cada byte em uma conexão TCP tem seu próprio número de seqüência de 32 bits. Quando o Internet começou, as linhas entre os roteadores eram principalmente linhas alugadas de 56 kbps, então um host explodindo em velocidade total levou mais de 1 semana para percorrer a sequência números. Em velocidades de rede modernas, os números de sequência podem ser consumidos em uma taxa alarmante, como veremos mais tarde. Números de sequência de 32 bits separados são carregados em pacotes para a posição da janela deslizante em uma direção e para reconhecer arestas na direção reversa, conforme discutido abaixo.

As entidades TCP de envio e recebimento trocam dados na forma de segmentos. Um **segmento TCP** consiste em um cabeçalho fixo de 20 bytes (mais uma parte opcional) seguido por zero ou mais bytes de dados. O software TCP decide o quanto grande os segmentos devem ser. Ele pode acumular dados de várias gravações em um segmento ou pode dividir os dados de uma gravação em vários segmentos. Dois limites restringem o tamanho do segmento. Primeiro, cada segmento, incluindo o cabeçalho TCP, deve caber no 65.515-byte IP payload. Em segundo lugar, cada link possui uma **MTU (Unidade Máxima de Transferência)**.

Cada segmento deve caber no MTU no remetente e no receptor para que possa ser enviado e recebidos em um único pacote não fragmentado. Na prática, o MTU é geralmente 1500 bytes (o tamanho da carga útil Ethernet) e, portanto, define o limite superior no tamanho do segmento.

No entanto, ainda é possível que os pacotes IP que transportam segmentos TCP sejam fragmentados ao passar por um caminho de rede para o qual algum link possui um pequeno MTU.

Se isso acontecer, degrada o desempenho e causa outros problemas (Kent e Mogul, 1987). Em vez disso, as implementações TCP modernas executam o **caminho MTU descoberta** usando a técnica descrita na RFC 1191 que descrevemos na Seç.

5.5.5. Esta técnica usa mensagens de erro ICMP para encontrar o menor MTU para qualquer link no caminho. O TCP então ajusta o tamanho do segmento para baixo para evitar fragmentação.

O protocolo básico usado por entidades TCP é o protocolo de janela deslizante com um tamanho dinâmico da janela. Quando um remetente transmite um segmento, ele também inicia um cronômetro.

Quando o segmento chega ao destino, a entidade TCP receptora envia de volta um segmento (com dados, se houver, e sem) contendo um reconhecimento número de ment igual ao próximo número de sequência que espera receber e o re-tamanho da janela principal. Se o temporizador do remetente desligar antes da confirmação for recebido, o remetente transmite o segmento novamente.

Embora este protocolo pareça simples, às vezes existem muitos insights sutis e saídas, que abordaremos a seguir. Os segmentos podem chegar fora de ordem, portanto, os bytes 3072–4095 podem chegar, mas não podem ser confirmados porque os bytes 2048–3071 têm não apareceu ainda. Os segmentos também podem ser atrasados tanto em trânsito que o remetente expira e os retransmite. As retransmissões podem incluir bytes diferentes

Página 581

SEC. 6,5

OS PROTOCOLOS DE TRANSPORTE DA INTERNET: TCP

557

alcances do que a transmissão original, exigindo uma administração cuidadosa para manter controle de quais bytes foram recebidos corretamente até o momento. No entanto, uma vez que cada byte no fluxo tem seu próprio deslocamento exclusivo, isso pode ser feito.

O TCP deve estar preparado para lidar com esses problemas e resolvê-los de forma eficaz maneira eficiente. Uma quantidade considerável de esforço foi canalizada para otimizar o desempenho

desempenho de fluxos TCP, mesmo em face de problemas de rede. Uma série de algoritmos usados por muitas implementações de TCP serão discutidos abaixo.

6.5.4 O cabeçalho do segmento TCP

A Figura 6-36 mostra o layout de um segmento TCP. Cada segmento começa com um cabeçalho de formato fixo de 20 bytes. O cabeçalho fixo pode ser seguido por uma opção de cabeçalho ções. Após as opções, se houver, até $65.535 - 20 - 20 = 65.495$ bytes de dados podem a seguir, onde os primeiros 20 se referem ao cabeçalho IP e os segundos ao cabeçalho TCP. Segmentos sem quaisquer dados são legais e são comumente usados para reconhecimento mentos e mensagens de controle.

32 bits
Porta de origem
Porto de destino
Número sequencial
Número de confirmação
TCP
cabeçalho
comprimento
você
R
G
UMA
C
K
P
S
H
R
S
T
S
Y
N
F
Eu

N
Tamanho da janela
Checksum
Ponteiro urgente
Opções (0 ou mais palavras de 32 bits)
Dados (opcional)
E
C
E
C
W
R

Figura 6-36. O cabeçalho TCP.

Vamos dissecar o cabeçalho TCP campo por campo. A *porta de origem e destino* campos de *porta* identificam os pontos finais locais da conexão. Uma porta TCP mais seu o endereço IP do host forma um ponto final exclusivo de 48 bits. A origem e o destino final pontos juntos identificam a conexão. Este identificador de conexão é chamado de **5 tupla** porque consiste em cinco informações: o protocolo (TCP), fonte IP e porta de origem e IP de destino e porta de destino.

Página 582

558

A CAMADA DE TRANSPORTE

INDIVÍDUO. 6

Os campos de *número de sequência* e *número de confirmação* realizam seus funções usuais. Observe que o último especifica o próximo byte esperado na ordem, não o último byte recebido corretamente. É um **reconhecimento cumulativo** porque resume os dados recebidos com um único número. Não vai além de perdido dados. Ambos têm 32 bits porque cada byte de dados é numerado em um fluxo TCP.

O *comprimento do cabeçalho TCP* informa quantas palavras de 32 bits estão contidas no TCP cabeçalho. Esta informação é necessária porque o campo *Opções* é de comprimento variável, então o cabeçalho também. Tecnicamente, este campo realmente indica o início dos dados dentro do segmento, medido em palavras de 32 bits, mas esse número é apenas o cabeçalho comprimento em palavras, então o efeito é o mesmo.

Em seguida, vem um campo de 4 bits que não é usado. O fato de que esses bits têm permanecido sem uso por 30 anos (como apenas 2 dos 6 bits originais reservados foram recuperado) é um testemunho de quão bem pensado é o TCP. Protocolos menores seriam precisaram desses bits para corrigir bugs no design original.

Agora vêm oito sinalizadores de 1 bit. *CWR* e *ECE* são usados para sinalizar congestionamento quando ECN (Notificação de Congestionamento Explícito) é usado, conforme especificado em RFC 3168.

ECE é definido para sinalizar um *ECN-Echo* para um remetente de TCP para dizer a ele para diminuir quando o receptor TCP obtém uma indicação de congestionamento da rede. *CWR* está definido para *janela de congestionamento de sinal reduzida* do emissor TCP para o receptor TCP, que sabe que o remetente diminuiu a velocidade e pode interromper o envio do *ECN-Echo*. Discutimos o papel do ECN no controle de congestionamento do TCP na Seção 6.5.10.

URG é definido como 1 se o *ponteiro Urgente* estiver em uso. O *ponteiro Urgente* é usado para indicar um deslocamento de byte do número da sequência atual em que os dados urgentes são a ser encontrado. Este recurso substitui mensagens de interrupção. Como mencionamos acima, esta facilidade é uma maneira básica de permitir que o remetente sinalize o receber sem envolver o próprio TCP no motivo da interrupção, mas é raramente usado.

O bit *ACK* é definido como 1 para indicar que o *número de confirmação* é válido. Esse é o caso de quase todos os pacotes. Se *ACK* for 0, o segmento não contém uma confirmação, portanto, o campo *Número de confirmação* é ignorado. O bit *PSH* indica dados PUSHed. O receptor é gentilmente solicitado para entregar os dados ao aplicativo na chegada e não armazená-los em buffer até um completo buffer foi recebido (o que poderia acontecer para eficiência). O bit *RST* é usado para reiniciar abruptamente uma conexão que se tornou confusa

devido a uma falha do host ou algum outro motivo. Também é usado para rejeitar um segmento inválido

mentar ou recusar uma tentativa de abrir uma conexão. Em geral, se você obtiver um segmento com o bit *RST ativado*, você tem um problema nas mãos.

O bit *SYN* é usado para estabelecer conexões. A solicitação de conexão tem *SYN = 1* e *ACK = 0* para indicar que o campo de confirmação de recebimento não é em uso. A resposta da conexão contém uma confirmação, no entanto, *SYN = 1* e *ACK = 1*. Em essência, o bit *SYN* é usado para denotar tanto CONNEC-PEDIDO DE INSTRUÇÃO e CONEXÃO ACEITA, com o bit *ACK* usado para distinguir palpite entre essas duas possibilidades.

Página 583

SEC. 6,5

OS PROTOCOLOS DE TRANSPORTE DA INTERNET: TCP

559

O bit *FIN* é usado para liberar uma conexão. Ele especifica que o remetente não tem mais dados para *transmitir*. No entanto, após fechar uma conexão, o processo de fechamento pode continuar a *receive*r dados indefinidamente. Ambos os segmentos *SYN* e *FIN* têm números de referência e, portanto, têm a garantia de serem processados na ordem correta. O controle de fluxo no TCP é feito usando uma janela deslizante de tamanho variável. O campo de *tamanho da janela* informa quantos bytes podem ser enviados a partir da confirmação de byte saliente. Um campo de *tamanho da janela* de 0 é legal e diz que os bytes até e incluem *ing número de confirmação - 1* foram recebidas, mas que o receptor tem não teve a chance de consumir os dados e não gostaria de mais dados para o momento, obrigado. O receptor pode mais tarde conceder permissão para enviar, transmitindo um segmento com o mesmo *número de confirmação* e um *tamanho da janela* diferente de zero campo.

Nos protocolos do cap. 3, confirmações de frames recebidos e por missão de enviar novos quadros foram amarrados. Isso foi consequência de um tamanho de janela fixo para cada protocolo. No TCP, reconhecimentos e permissão para enviar dados adicionais são completamente dissociados. Com efeito, um receptor pode dizer: "I recebi bytes até *k*, mas não quero mais nenhum agora, obrigado você." Este desacoplamento (na verdade, uma janela de tamanho variável) dá flexibilidade adicional ity. Vamos estudá-lo em detalhes abaixo.

Um *checksum* também é fornecido para maior confiabilidade. Ele faz a soma de verificação do cabeçalho,

os dados e um pseudo-cabeçalho conceitual exatamente da mesma maneira que UDP, exceto que o pseudoheader tem o número do protocolo para TCP (6) e a soma de verificação é obrigatório. Por favor, consulte a seção 6.4.1 para detalhes.

O campo *Opções* fornece uma maneira de adicionar recursos extras não abrangidos pelo cabeçalho regular. Muitas opções foram definidas e várias são comumente usadas.

As opções são de comprimento variável, preencha um múltiplo de 32 bits usando preenchimento com

zeros, e pode se estender até 40 bytes para acomodar o cabeçalho TCP mais longo que pode ser especificado. Algumas opções são realizadas quando uma conexão é estabelecida para ne-iniciar ou informar o outro lado das capacidades. Outras opções são carregadas na embalagem ets durante a vida útil da conexão. Cada opção tem um Tipo-Comprimento-Valor codificação.

Uma opção amplamente utilizada é aquela que permite que cada host especifique o **MSS** (**Tamanho máximo do segmento**) está disposto a aceitar. Usar grandes segmentos é mais eficiente do que usar pequenos porque o cabeçalho de 20 bytes pode ser amortizado durante mais dados, mas pequenos hosts podem não ser capazes de lidar com grandes segmentos. Durante o con-

configuração da conexão, cada lado pode anunciar seu máximo e ver o de seu parceiro. Se um hospedeiro

não usa essa opção, o padrão é uma carga útil de 536 bytes. Todos os hosts da Internet são

necessário para aceitar segmentos TCP de $536 + 20 = 556$ bytes. O segmento máximo o tamanho do movimento nas duas direções não precisa ser o mesmo.

Para linhas com alta largura de banda, alto atraso ou ambos, a janela de 64 KB corresponde pular para um campo de 16 bits é um problema. Por exemplo, em uma linha OC-12 (de aproximadamente

600 Mbps), leva menos de 1 ms para gerar uma janela completa de 64 KB. Se o atraso de propagação de ida e volta é de 50 ms (o que é típico para um

Página 584

560

A CAMADA DE TRANSPORTE INDIVÍDUO. 6

fibra), o remetente ficará inativo por mais de 98% do tempo esperando pelo reconhecimentos. Um tamanho de janela maior permitiria ao remetente continuar a enviar dados.

A opção de **escala da janela** permite que o remetente e o destinatário negociem uma janela fator de escala no início de uma conexão. Ambos os lados usam o fator de escala para deslocar o Campo de *tamanho de janela* de até 14 bits à esquerda, permitindo janelas de até 2^{30} bytes. A maioria das implementações TCP oferece suporte a essa opção.

A opção **timestamp** carrega um timestamp enviado pelo remetente e ecoado por o receptor. Ele está incluído em cada pacote, uma vez que seu uso é estabelecido durante a configuração de conexão, e usado para calcular amostras de tempo de ida e volta que são usadas para estimar

companheiro quando um pacote foi perdido. Também é usado como uma extensão lógica do 32- número de sequência de bits. Em uma conexão rápida, o número de sequência pode quebrar rapidamente, levando a uma possível confusão entre dados antigos e novos. o

Os descartes do esquema PAWS (proteção contra números de sequência embrulhada) dividir segmentos com carimbos de data / hora antigos para evitar esse problema.

Finalmente, a opção **SACK (reconhecimento seletivo)** permite que um receptor diga um remetente os intervalos de números de sequência que recebeu. Suplementa o Número de confirmação e é usado depois que um pacote foi perdido, mas subsequente (ou dados duplicados) chegaram. Os novos dados não são refletidos pelo Reconhecimento campo de *número de ment* no cabeçalho porque esse campo fornece apenas o próximo na ordem byte que é esperado. Com o SACK, o remetente está explicitamente ciente de quais dados o receptor tem e, portanto, pode determinar quais dados devem ser retransmitidos. SACO é definido no RFC 2108 e no RFC 2883 e é cada vez mais usado. Nós descrevemos o uso de SACK junto com controle de congestionamento na Seção 6.5.10.

6.5.5 Estabelecimento de Conexão TCP

As conexões são estabelecidas no TCP por meio do handshake de três vias dirigido na Seç. 6.2.2. Para estabelecer uma conexão, um lado, digamos, o servidor, passe aguarda ativamente por uma conexão de entrada executando LISTEN e ACCEPT primitivas nessa ordem, especificando uma fonte específica ou ninguém em particular. O outro lado, digamos, o cliente, executa uma primitiva CONNECT , especificando o Endereço IP e porta à qual deseja se conectar, o tamanho máximo do segmento TCP está disposto a aceitar e, opcionalmente, alguns dados do usuário (por exemplo, uma senha). O CON- A primitiva NECT envia um segmento TCP com o bit SYN ligado e o bit ACK desligado e aguarda uma resposta.

Quando este segmento chega ao destino, a entidade TCP verifica se veja se existe um processo que fez um LISTEN na porta dada em *Destination campo portuário* . Caso contrário, ele envia uma resposta com o bit RST ativado para rejeitar a conexão.

Se algum processo estiver escutando a porta, esse processo receberá o Segmento TCP. Ele pode aceitar ou rejeitar a conexão. Se aceitar, um ac- segmento de conhecimento é enviado de volta. A sequência de segmentos TCP enviados no o caso normal é mostrado na Figura 6.37 (a). Observe que um segmento SYN consome 1 byte de espaço de sequência para que possa ser reconhecido sem ambigüidade.

SEC. 6,5
OS PROTOCOLOS DE TRANSPORTE DA INTERNET: TCP

561

```

Tempo
Host 1
Host 2
SYN (SEQ = y, ACK = x + 1)
SYN (SEQ = x)
(SEQ = x + 1, ACK = y + 1)
Host 1
Host 2
SYN (SEQ = y, ACK = x + 1)
SYN (SEQ = x)
SYN (SEQ = y)
SYN (SEQ = x, ACK = y + 1)
(uma)
(b)

```

Figura 6-37. (a) Estabelecimento de conexão TCP no caso normal. (b) Simultâneo estabelecimento de conexão instantânea em ambos os lados.

No caso de dois hosts tentarem estabelecer uma conexão simultaneamente entre os mesmos dois soquetes, a sequência de eventos é ilustrada na Fig. 6-37 (b). O resultado desses eventos é que apenas uma conexão é estabelecida, não dois, porque as conexões são identificadas por seus pontos finais. Se a primeira configuração voltar resulta em uma conexão identificada por (x, y) e o segundo também, apenas um a entrada na tabela é feita, a saber, para (x, y) .

Lembre-se de que o número de sequência inicial escolhido por cada host deve circular lentamente, em vez de ser uma constante, como 0. Esta regra é para proteger contra atrasos pacotes duplicados, como discutimos na Seção 6.2.2. Originalmente, isso foi realizado com um esquema baseado em relógio no qual o relógio marca a cada 4 µseg.

No entanto, uma vulnerabilidade com a implementação do handshake de três vias é que o processo de escuta deve lembrar o seu número de sequência, logo que responde com seu próprio segmento SYN. Isso significa que um remetente malicioso pode bloquear recursos em um host enviando um fluxo de segmentos SYN e nunca seguindo para completar a conexão. Este ataque é chamado de **inundação SYN**, e aleijou muitos Servidores da Web na década de 1990.

Uma maneira de se defender contra esse ataque é usar **cookies SYN**. Ao invés de lembrando o número de sequência, um host escolhe um gerado criptograficamente número de sequência, coloca-o no segmento de saída e esquece-o. Se as três vias o handshake for concluído, este número de sequência (mais 1) será retornado ao host. Ele pode, então, regenerar o número de sequência correto executando o mesmo cripto-função gráfica, desde que as entradas para essa função sejam conhecidas, por exemplo, o endereço IP e a porta de outro host e um segredo local. Este procedimento permite que o host para verificar se um número de sequência reconhecido está correto sem ter que

562

A CAMADA DE TRANSPORTE
INDIVÍDUO. 6

Lembre-se do número de sequência separadamente. Existem algumas ressalvas, como a incapacidade de lidar com as opções TCP, então os cookies SYN podem ser usados apenas quando o host está sujeito a uma inundação de SYN. No entanto, eles são uma reviravolta interessante na conexão estabelecimento. Para obter mais informações, consulte RFC 4987 e Lemon (2002).

6.5.6 Liberação de Conexão TCP

Embora as conexões TCP sejam full duplex, para entender como as conexões são lançadas, é melhor pensar neles como um par de conexões simplex. Cada simplex a conexão é liberada independentemente de seu irmão. Para liberar uma conexão, uma parte pode enviar um segmento TCP com o conjunto de bits FIN, o que significa que ele não tem mais dados para transmitir. Quando o FIN é reconhecido, essa direção é desligada para novos dados. Os dados podem continuar a fluir indefinidamente na outra direção, sempre. Quando ambas as direções forem desligadas, a conexão será encerrada.

Normalmente, quatro segmentos TCP são necessários para liberar uma conexão: um *FIN* e um *ACK* para cada direção. No entanto, é possível para o primeiro *ACK* e o segundo *FIN* estar contido no mesmo segmento, reduzindo a contagem total para três. Tal como acontece com as chamadas telefônicas em que ambas as pessoas se despedem e desligam o telefone simultaneamente, ambas as extremidades de uma conexão TCP podem enviar segmentos *FIN* em o mesmo tempo. Cada um deles é reconhecido da maneira usual, e a conexão é encerrada. Não há, de fato, nenhuma diferença essencial entre os dois hosts liberando sequencialmente ou simultaneamente.

Para evitar o problema de dois exércitos (discutido na Seção 6.2.3), são usados temporizadores. Se um resposta a um *FIN* não está disponível dentro de dois tempos de vida máximos do pacote, o remetente do *FIN* libera a conexão. O outro lado irá eventualmente notar que parece que ninguém mais está ouvindo e também vai desligar o tempo. Enquanto esta solução não é perfeita, dado o fato de que uma solução perfeita é teoricamente impossível, terá que servir. Na prática, raramente surgem problemas.

6.5.7 Modelagem de Gerenciamento de Conexão TCP

As etapas necessárias para estabelecer e liberar conexões podem ser representadas em um máquina de estados finitos com os 11 estados listados na Figura 6.38. Em cada estado, certo eventos são legais. Quando um evento legal acontece, algumas medidas podem ser tomadas. Se algum

outro evento acontece, um erro é relatado.

Cada conexão começa no estado *CLOSED*. Ele sai desse estado quando o faz ou uma abertura passiva (*LISTEN*) ou uma abertura ativa (*CONNECT*). Se o outro lado faz o oposto, uma conexão é estabelecida e o estado torna-se *ESTABLISHED*. A liberação da conexão pode ser iniciada por qualquer um dos lados. Quando é com completo, o estado retorna para *FECHADO*.

A própria máquina de estados finitos é mostrada na Figura 6.39. O caso comum de um cliente conectadoativamente a um servidor passivo é mostrado com linhas pesadas - sólido para o cliente, pontuado para o servidor. As linhas da face de luz são sequências de eventos incomuns.

Página 587

SEC. 6,5
OS PROTOCOLOS DE TRANSPORTE DA INTERNET: TCP

563

Estado

Descrição

FECHADAS

Nenhuma conexão está ativa ou pendente

OUÇO

O servidor está esperando por uma chamada

SYN RCVD

Uma solicitação de conexão chegou; espere por ACK

SYN SENT

O aplicativo começou a abrir uma conexão

ESTABELECIDO

O estado normal de transferência de dados

FIN WAIT 1

O aplicativo disse que foi concluído

FIN WAIT 2

O outro lado concordou em liberar

TEMPO DE ESPERA

Espere que todos os pacotes morram

ENCERRAMENTO

Ambos os lados tentaram fechar simultaneamente

FECHAR ESPERA

O outro lado iniciou uma liberação

ÚLTIMO ACK

Espere que todos os pacotes morram

Figura 6-38. Os estados usados na máquina de estado finito de gerenciamento de conexão TCP.

Cada linha na Figura 6-39 é marcada por um par *evento / ação*. O evento pode ser

uma chamada de sistema iniciada pelo usuário (`CONNECT` , `LISTEN` , `SEND` ou `CLOSE`), uma chegada de segmento

(`SYN` , `FIN` , `ACK` ou `RST`), ou, em um caso, um tempo limite de duas vezes o pacote máximo tempo de vida. A ação é o envio de um segmento de controle (`SYN` , `FIN` ou `RST`) ou nada, indicado por `-`. Os comentários são mostrados entre parênteses.

Pode-se entender melhor o diagrama seguindo primeiro o caminho de um cliente (a linha sólida grossa) e, mais tarde, seguindo o caminho de um servidor (a linha tracejada grossa). Quando um programa de aplicação na máquina cliente emite um `CONNECT` rebusca, a entidade TCP local cria um registro de conexão, marca-o como estando no Estado `SYN SENT` e dispara um segmento `SYN`. Observe que muitas conexões podem ser aberto (ou sendo aberto) ao mesmo tempo em nome de vários aplicativos, então o estado é por conexão e registrado no registro de conexão. Quando o `SYN + ACK` chega, TCP envia o `ACK` final do handshake de três vias e muda para o estado `ESTABLISHED`. Os dados agora podem ser enviados e recebidos. Quando um aplicativo é finalizado, ele executa uma primitiva `CLOSE` , o que causa a entidade TCP local para enviar um segmento `FIN` e esperar pelo `ACK` correspondente (caixa tracejada marcada " fechamento ativo "). Quando o `ACK` chega, uma transição é feita para o estado `FIN WAIT 2` e uma direção da conexão é fechada. Quando o outro lado fecha, também, entra um `FIN` , que é reconhecido. Agora ambos os lados estão fechados, mas o TCP espera um tempo igual a duas vezes o tempo de vida máximo do pacote para garantir que todos os pacotes da conexão morreram, apenas no caso do acidente o conhecimento foi perdido. Quando o cronômetro apaga, o TCP exclui a conexão registro.

Agora, vamos examinar o gerenciamento de conexão do ponto de vista do servidor. O servidor faz um `LISTEN` e se estabelece para ver quem aparece. Quando um `SYN`

Página 588

564

A CAMADA DE TRANSPORTE

INDIVÍDUO. 6

FECHADAS

OUÇO

ESTABELECIDO

ENCERRAMENTO

FECHAR

ESPERAR

(Começar)

`CONNECT / SYN` (Etapa 1 do handshake de três vias)

ESCUTE / -

`SYN / SYN + ACK`

`SYN`

`RCVD`

`FIN`

`ESPERE 1`

`TEMPO`

`ESPERAR`

`ÚLTIMO`

`ACK`

`FIN`

`ESPERE 2`

`SYN`

`ENVIEI`

`RST / -`

`ACK / -`

(Fechamento ativo)

`FIN / ACK`

`FIN + ACK / ACK`

`FIN / ACK`

`ACK / -`

`ACK / -`

`ACK / -`

`SEND / SYN`

`SYN / SYN + ACK`

(abertura simultânea)

(Estado de transferência de dados)

`SYN + ACK / ACK`

(Etapa 3 do handshake de 3 vias)

`CLOSE / FIN`

`CLOSE / FIN`

FIN / ACK
FECHAR / -
FECHAR / -
CLOSE / FIN
FECHADAS
(Fechamento passivo)
(Tempo esgotado/)
(Voltar para o início)
(Passo 2
do handshake de 3 vias)

Figura 6-39. Máquina de estado finito de gerenciamento de conexão TCP. O pesado a linha sólida é o caminho normal para um cliente. A linha tracejada pesada é o normal caminho para um servidor. As linhas de luz são eventos incomuns. Cada transição é rotulada com o evento que o está causando e a ação resultante dele, separados por uma barra.
entra, é confirmado e o servidor vai para o estado *SYN RCVD*. Quando o *SYN* do servidor é ele próprio reconhecido, o handshake de três vias está completo e o servidor vai para o estado *ESTABLISHED*. A transferência de dados pode agora ocorrer. Quando o cliente termina de transmitir seus dados, ele faz um *CLOSE*, o que causa um *FIN* para chegar ao servidor (caixa tracejada marcada como "fechamento passivo"). O servidor é então sinalizado. Quando ele também fecha, um *FIN* é enviado ao cliente. Quando o

Página 589

SEC. 6,5
OS PROTOCOLOS DE TRANSPORTE DA INTERNET: TCP

565

a confirmação do cliente aparece, o servidor libera a conexão e exclui o registro de conexão.

6.5.8 Janela Deslizante TCP

Como mencionado anteriormente, o gerenciamento de janela no TCP desacopla os problemas de confirmação do recebimento correto de segmentos e buffer de receptor alocação. Por exemplo, suponha que o receptor tenha um buffer de 4.096 bytes, conforme mostrado em Fig. 6-40. Se o remetente transmitir um segmento de 2.048 bytes que é recebido corretamente, o receptor reconhecerá o segmento. No entanto, como agora tem apenas 2.048 bytes de espaço do buffer (até que o aplicativo remova alguns dados do buffer), anunciará uma janela de 2048 começando no próximo byte esperado.

Inscrição
faz um 2-KB
escrever
Inscrição
faz um 2-KB
escrever
Inscrição
lê 2 KB
Remetente é
bloqueado
Remetente pode
enviar até 2 KB
Do receptor
amortecedor
0
4 KB
2 KB
2 KB
Vazio
Cheio
2 KBSEQ = 0
2 KB
SEQ = 2048
ACK = 2048 WIN = 2048
ACK = 4096 WIN = 0
ACK = 4096 WIN = 2048
2 KB
1 KB
Remetente
Receptor

Figura 6-40. Gerenciamento de janelas em TCP.

Agora o remetente transmite outros 2.048 bytes, que são confirmados, mas a janela anunciada é de tamanho 0. O remetente deve parar até o aplicativo

566

A CAMADA DE TRANSPORTE
INDIVÍDUO. 6

processo no host receptor removeu alguns dados do buffer, no qual vez que o TCP pode anunciar uma janela maior e mais dados podem ser enviados. Quando a janela é 0, o remetente pode normalmente não enviar segmentos, com dois exceções. Primeiro, dados urgentes podem ser enviados, por exemplo, para permitir que o usuário mate o processo em execução na máquina remota. Em segundo lugar, o remetente pode enviar um 1 byte segmento para forçar o receptor a anunciar novamente o próximo byte esperado e a vitória tamanho baixo. Este pacote é denominado **janela de teste**. O padrão TCP explicitamente fornece esta opção para evitar deadlock se uma atualização de janela for perdida. Os remetentes não são obrigados a transmitir dados assim que eles chegam do aplicativo plicatura. Nem os destinatários são obrigados a enviar reconhecimentos assim que possível. Por exemplo, na Figura 6.40, quando os primeiros 2 KB de dados chegaram, TCP, sabendo que ele tinha uma janela de 4 KB, estaria completamente correto em apenas armazenar os dados em buffer até que outros 2 KB entrem, para poder transmitir um segmento com uma carga útil de 4 KB. Essa liberdade pode ser usada para melhorar o desempenho. Considere uma conexão a um terminal remoto, por exemplo, usando SSH ou telnet, que reage a cada pressionamento de tecla. No pior caso, sempre que um personagem chega a entidade TCP de envio, o TCP cria um segmento TCP de 21 bytes, que dá ao IP para enviar como um datagrama IP de 41 bytes. No lado receptor, o TCP imediatamente envia um Confirmação de 40 bytes (20 bytes de cabeçalho TCP e 20 bytes de cabeçalho IP). Mais tarde, quando o terminal remoto leu o byte, o TCP envia uma atualização da janela, movendo a janela 1 byte para a direita. Este pacote também tem 40 bytes. Finalmente, quando o terminal remoto processou o caractere, ele ecoa o caractere para local exibir usando um pacote de 41 bytes. Ao todo, 162 bytes de largura de banda são usados e quatro segmentos são enviados para cada caractere digitado. Quando a largura de banda é escassa, este método od de fazer negócios não é deseável. Uma abordagem que muitas implementações TCP usam para otimizar esta situação é chamado de **reconhecimentos atrasados**. A ideia é atrasar os reconhecimentos e atualizações de janela por até 500 ms, na esperança de adquirir alguns dados em qual pegar uma carona grátis. Assumindo que o terminal ecoa em 500 ms, apenas um pacote de 41 bytes agora precisa ser enviado de volta pelo lado remoto, cortando o pacote contagem e uso de largura de banda pela metade. Embora as confirmações atrasadas reduzam a carga colocada na rede pelo receptor, um remetente que envia vários pacotes curtos (por exemplo, pacotes de 41 bytes contendo 1 byte de dados) ainda está operando de forma ineficiente. Uma maneira de reduzir isso o uso é conhecido como **algoritmo de Nagle** (Nagle, 1984). O que Nagle sugeriu é simples: quando os dados chegam ao remetente em pequenos pedaços, basta enviar o primeiro e armazene todo o resto até que a primeira parte seja confirmada. Em seguida, envie todos os dados em buffer em um segmento TCP e começar a armazenar em buffer novamente até o próximo segmento é reconhecido. Ou seja, apenas um pacote curto pode estar pendente a qualquer momento. Se muitos dados forem enviados pelo aplicativo em um tempo de ida e volta, o de Nagle algoritmo irá colocar as muitas peças em um segmento, reduzindo significativamente a banda largura usada. O algoritmo também diz que um novo segmento deve ser enviado se dados suficientes fluíram para preencher um segmento máximo.

SEC. 6,5

OS PROTOCOLOS DE TRANSPORTE DA INTERNET: TCP

567

O algoritmo de Nagle é amplamente usado por implementações TCP, mas às vezes quando é melhor desativá-lo. Em particular, em jogos interativos que são atropelados

na Internet, os jogadores geralmente desejam um fluxo rápido de pacotes de atualização curtos. Reunir as atualizações para enviá-las em rajadas faz com que o jogo responda de forma irregular, o que torna os usuários insatisfeitos. Um problema mais sutil é que o algoritmo de Nagle às vezes pode interagir com confirmações atrasadas para causar uma impasse: o receptor aguarda os dados para pegar uma confirmação, e o remetente aguarda a confirmação para enviar mais dados. Esta interação pode atrasar o download de páginas da web. Por causa desses problemas, o algoritmo de Nagle O ritmo pode ser desabilitado (o que é chamado de opção *TCP NODELAY*). Mogul e Minshall (2001) discute essa e outras soluções.

Outro problema que pode degradar o desempenho do TCP é a **sincronização de janela boba**

Drome (Clark, 1982). Este problema ocorre quando os dados são passados para o envio

Entidade TCP em grandes blocos, mas um aplicativo interativo no lado receptor

lê dados apenas 1 byte de cada vez. Para ver o problema, veja a Figura 6-41. Inicialmente,

o buffer TCP no lado receptor está cheio (ou seja, tem uma janela de tamanho 0) e o

remetente sabe disso. Em seguida, o aplicativo interativo lê um caractere do

Fluxo TCP. Esta ação deixa o TCP receptor feliz, então ele envia uma janela

atualize para o remetente dizendo que pode enviar 1 byte. O remetente obriga

e envia 1 byte. O buffer agora está cheio, então o receptor reconhece o 1 byte

segmento e define a janela para 0. Este comportamento pode durar para sempre.

A solução de Clark é evitar que o receptor envie uma atualização de janela para 1

byte. Em vez disso, ele é forçado a esperar até que tenha uma quantidade decente de espaço

disponível

e anuncie isso. Especificamente, o receptor não deve enviar uma janela

atualizar até que possa lidar com o tamanho máximo de segmento anunciado quando o

conexão foi estabelecida ou até que seu buffer esteja meio vazio, o que for menor.

Além disso, o remetente também pode ajudar, não enviando segmentos minúsculos. Em vez disso deve esperar até que possa enviar um segmento completo, ou pelo menos um contendo metade do tamanho do buffer do receptor.

O algoritmo de Nagle e a solução de Clark para a síndrome da janela boba são complementar. Nagle estava tentando resolver o problema causado pelo aplicativo de envio plicação entregando dados ao TCP um byte de cada vez. Clark estava tentando resolver o problema do aplicativo de recepção sugando os dados do TCP um byte em um Tempo. Ambas as soluções são válidas e podem funcionar juntas. O objetivo é para o remetente não enviar pequenos segmentos e o receptor não pedir por eles.

O TCP receptor pode ir mais longe na melhoria do desempenho do que apenas fazer atualizações de janela em unidades grandes. Como o TCP de envio, ele também pode armazenar dados em buffer,

pode bloquear um pedido READ do aplicativo até que ele tenha um grande bloco de dados para isso. Isso reduz o número de chamadas ao TCP (e a sobrecarga). Isso também

aumenta o tempo de resposta, mas para aplicativos não interativos, como transferência de arquivos, a eficiência pode ser mais importante do que o tempo de resposta a solicitações individuais.

Outro problema que o receptor deve lidar é que os segmentos podem chegar fora do ordem. O receptor armazenará os dados em buffer até que eles possam ser passados para o aplicativo

568

A CAMADA DE TRANSPORTE

INDIVÍDUO. 6

O aplicativo lê 1 byte

Segmento de atualização de janela enviado

Novo byte chega

Cabeçalho

Cabeçalho

O buffer do receptor está cheio

O buffer do receptor está cheio

Espaço para mais um byte

1 byte

Figura 6-41. Síndrome da janela boba.

em ordem. Na verdade, nada de ruim aconteceria se os segmentos fora de ordem fossem destruídos cardados, uma vez que eventualmente seriam retransmitidos pelo remetente, mas seriam

ser um desperdício.

Agradecimentos podem ser enviados somente quando todos os dados até o byte são confirmados. Segundo foram recebidos. Isso é chamado de **confirmação cumulativa**. E se o receptor obtém os segmentos 0, 1, 2, 4, 5, 6 e 7, ele pode reconhecer tudo e incluindo o último byte no segmento 2. Quando o remetente atinge o tempo limite, ele então retransmite o segmento 3. Como o receptor armazenou os segmentos 4 a 7, após o recebimento do segmento 3 pode reconhecer todos os bytes até o final do segmento 7.

6.5.9 Gerenciamento de Timer TCP

O TCP usa vários temporizadores (pelo menos conceitualmente) para fazer seu trabalho. O mais importante destes é o **RTO (Retransmission TimeOut)**. Quando um segmento é enviado, um temporizador de retransmissão é iniciado. Se o segmento for reconhecido antes do cronômetro expira, o cronômetro é interrompido. Se, por outro lado, o cronômetro dispara antes a confirmação chega, o segmento é retransmitido (e o temporizador é começado novamente). A questão que se coloca é: quanto tempo deve ser o tempo limite? Este problema é muito mais difícil na camada de transporte do que no enlace de dados protocolos como 802.11. Neste último caso, o atraso esperado é medido em

Página 593

SEC. 6,5

OS PROTOCOLOS DE TRANSPORTE DA INTERNET: TCP

569

microssegundos e é altamente previsível (ou seja, tem uma variação baixa), então o cronômetro pode ser definido para desligar um pouco depois que o reconhecimento é esperado, conforme mostrado em

Fig. 6-42 (a). Uma vez que as confirmações raramente são atrasadas na camada de enlace de dados (devido à falta de congestionamento), a ausência de um reconhecimento no esperado tempo geralmente significa que o quadro ou a confirmação foi perdido.

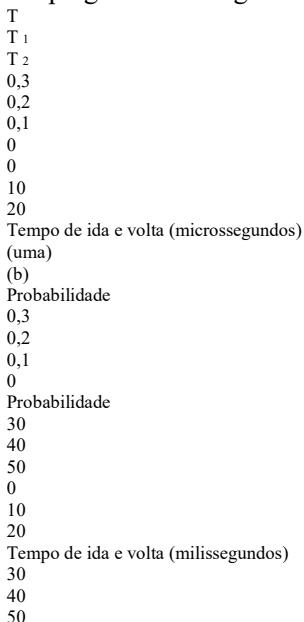


Figura 6-42. (a) Densidade de probabilidade de tempos de chegada de confirmação no camada de enlace de dados. (b) Densidade de probabilidade de tempos de chegada de confirmação para TCP.

O TCP se depara com um ambiente radicalmente diferente. A densidade de probabilidade função pelo tempo que leva para uma confirmação de TCP retornar parece mais parecido com a Fig. 6-42 (b) do que com a Fig. 6-42 (a). É maior e mais variável. Determinar o tempo de ida e volta até o destino é complicado. Mesmo quando é conhecido, decidir sobre o intervalo de tempo limite também é difícil. Se o tempo limite for muito curto, por exemplo, T_1 na Fig. 6-42 (b), retransmissões desnecessárias irão ocorrer, o entupimento Internet com pacotes inúteis. Se for definido muito longo (por exemplo, T_2), o desempenho será prejudicado

devido ao longo atraso de retransmissão sempre que um pacote é perdido. Além disso, o

média e variância da distribuição de chegada de confirmação podem mudar rapidamente dentro de alguns segundos, conforme o congestionamento aumenta ou é resolvido. A solução é usar um algoritmo dinâmico que adapta constantemente o tempo limite intervalo, com base em medições contínuas de desempenho da rede. O algoritmo geralmente usado pelo TCP é devido a Jacobson (1988) e funciona da seguinte maneira. Para cada conexão, o TCP mantém uma variável, *SRTT* (Smoothed Round-Trip Time), essa é a melhor estimativa atual do tempo de ida e volta para o destino em questão. Quando um segmento é enviado, um cronômetro é iniciado, tanto para ver quanto tempo o conhecimento leva e também para disparar uma retransmissão se demorar muito. E se

570

A CAMADA DE TRANSPORTE INDIVÍDUO. 6

a confirmação volta antes que o tempo expire, o TCP mede por quanto tempo o reconhecimento levou, digamos, *R*. Em seguida, atualiza *SRTT* de acordo com a fórmula $SRTT = \alpha SRTT + (1 - \alpha) R$

onde α é um fator de suavização que determina a rapidez com que os valores antigos são para-obtido. Normalmente, $\alpha = 7 / 8$. Este tipo de fórmula é um EWMA (Exponencialmente

Média móvel ponderada) ou filtro passa-baixa que descarta o ruído nas amostras.

Mesmo com um bom valor de *SRTT*, escolher um tempo limite de retransmissão adequado é um assunto não trivial. As implementações iniciais do TCP usaram $2 \times RTT$, mas a experiência mostrou que um valor constante era muito inflexível porque falhou em responder quando a variação aumentou. Em particular, modelos de enfileiramento aleatórios (ou seja, Poisson) o tráfego prevê que quando a carga se aproxima da capacidade, o atraso se torna grande e altamente variável. Isso pode levar ao disparo do temporizador de retransmissão e a uma cópia do pacote sendo retransmitido, embora o pacote original ainda esteja em trânsito no rede. É muito mais provável que aconteça em condições de alta carga, que é o pior momento para enviar pacotes adicionais para a rede.

Para corrigir esse problema, Jacobson propôs tornar o valor de tempo limite sensível a a variação nos tempos de ida e volta, bem como o tempo de ida e volta suavizado. Isto mudança requer o acompanhamento de outra variável suavizada, *RTTVar* (Round-Trip Time VARiation) que é atualizado usando a fórmula

$$RTTVar = \beta RTTVar + (1 - \beta) | SRTT - R |$$

Este é um EWMA como antes, e normalmente $\beta = 3 / 4$. O tempo limite de retransmissão, *RTT*, está definido para ser $RTT = SRTT + 4 \times RTTVar$

A escolha do fator 4 é um tanto arbitrária, mas a multiplicação por 4 pode ser feito com um único turno, e menos de 1% de todos os pacotes vêm em mais de quatro desvios padrão tardios. Observe que *RTTVar* não é exatamente igual ao padrão desvio (é realmente o desvio médio), mas é próximo o suficiente na prática.

O artigo de Jacobson está cheio de truques inteligentes para calcular tempos limite usando apenas números inteiros

adiciona, subtrai e desloca. Esta economia não é necessária para hosts modernos, mas tornou-se parte da cultura que permite que o TCP seja executado em todos os tipos de dispositivos, desde supercomputadores até dispositivos minúsculos. Até agora, ninguém colocou em um RFID chip, mas algum dia? Quem sabe.

Mais detalhes sobre como calcular esse tempo limite, incluindo as configurações iniciais do variáveis, são fornecidas no RFC 2988. O temporizador de retransmissão também é mantido em um mini-mãe de 1 segundo, independentemente das estimativas. Este é um valor conservador escolhido para evitar retransmissões espúrias com base em medições (Allman e Paxson, 1999).

Um problema que ocorre com a coleta de amostras, *R*, do tempo de ida e volta é o que fazer quando um segmento atinge o tempo limite e é enviado novamente. Quando o reconhecimento

edgement entra, não está claro se o reconhecimento se refere ao primeiro

SEC. 6,5

OS PROTOCOLOS DE TRANSPORTE DA INTERNET: TCP

571

transmissão ou posterior. O palpito errado pode contaminar seriamente o tempo limite de transmissão. Phil Karn descobriu esse problema da maneira mais difícil. Karn é um entusiasta de rádio amador interessado em transmitir pacotes TCP / IP por radioamador rádio, um meio notoriamente não confiável. Ele fez uma proposta simples: não levante estimativas de data em quaisquer segmentos que tenham sido retransmitidos. Além disso, o tempo limite é dobrado em cada retransmissão sucessiva até que os segmentos obtenham pela primeira vez. Esta correção é chamada **de algoritmo de Karn** (Karn e Partridge, 1987). A maioria das implementações de TCP o usa.

O temporizador de retransmissão não é o único temporizador que o TCP usa. Um segundo cronômetro é

o **cronômetro de persistência**. Ele foi projetado para evitar o seguinte deadlock. Lá-
ceiver envia uma confirmação com um tamanho de janela de 0, dizendo ao remetente para esperar. Mais tarde, o receptor atualiza a janela, mas o pacote com a atualização é perdido. Agora, o remetente e o receptor estão esperando que o outro faça alguma coisa. Quando o temporizador de persistência dispara, o remetente transmite uma sonda para o receptor. A resposta ao probe fornece o tamanho da janela. Se ainda for 0, o per-
o temporizador de emergência é definido novamente e o ciclo se repete. Se for diferente de zero, os dados agora podem ser
enviei.

Um terceiro cronômetro que algumas implementações usam é o **cronômetro de manutenção de atividade**. Quando um

a conexão ficou inativa por um longo tempo, o cronômetro de manutenção de atividade pode desligar para causar

um lado para verificar se o outro lado ainda está lá. Se não responder, o con-
exão é encerrada. Este recurso é controverso porque adiciona sobrecarga e
pode encerrar uma conexão de outra forma saudável devido a uma partição de rede transitória
ção.

O último temporizador usado em cada conexão TCP é aquele usado no *TIME*
Estado *ESPERA* ao fechar. Ele funciona por duas vezes a vida útil máxima do pacote para fazer
Certifique-se de que, quando uma conexão é fechada, todos os pacotes criados por ela morrem.

6.5.10 Controle de Congestionamento TCP

Deixamos uma das funções-chave do TCP para o final: controle de congestionamento.
Quando a carga oferecida a qualquer rede é maior do que ela pode suportar, o congestionamento acumula. A Internet não é exceção. A camada de rede detecta congestionamento quando as filas ficam grandes nos roteadores e tenta gerenciá-las, mesmo que apenas diminuindo pacotes. Cabe à camada de transporte receber feedback de congestionamento da camada de rede e diminui a taxa de tráfego que está enviando para a rede.

Na Internet, o TCP desempenha o papel principal no controle do congestionamento, bem como o papel principal no transporte confiável. É por isso que é um protocolo tão especial.

Cobrimos a situação geral de controle de congestionamento na Seção 6.3. Uma chave aprendido foi que um protocolo de transporte usando um AIMD (Additive Aumento Multi-
redução plicativa) lei de controle em resposta a sinais binários de congestionamento da rede convergiria para uma alocação de largura de banda justa e eficiente. TCP com
controle de gestão é baseado na implementação desta abordagem usando uma janela e com
perda de pacotes como o sinal binário. Para fazer isso, o TCP mantém uma **janela de congestionamento**

572

A CAMADA DE TRANSPORTE

INDIVÍDUO. 6

cujo tamanho é o número de bytes que o remetente pode ter na rede a qualquer momento.

A taxa correspondente é o tamanho da janela dividido pelo tempo de ida e volta da conexão. O TCP ajusta o tamanho da janela de acordo com a regra AIMD.

Lembre-se de que a janela de congestionamento é mantida, *além* do fluxo de controle de janela trol, que especifica o número de bytes que o receptor pode armazenar em buffer.

Ambas as janelas são rastreadas em paralelo, e o número de bytes que podem ser enviados é menor das duas janelas. Assim, a janela efetiva é menor de

o que o emissor pensa está correto e o que o receptor pensa está correto. Leva dois para dançar o tango. O TCP irá parar de enviar dados se o congestionamento ou o fluxo de janela trol está temporariamente cheio. Se o destinatário disser "enviar 64 KB", mas o remetente sabe que rajadas de mais de 32 KB obstruem a rede, ele enviará 32 KB. Em por outro lado, se o receptor disser "enviar 64 KB" e o remetente souber que rajadas de até 128 KB passam sem esforço, ele enviará o total de 64 KB questionado. A janela de controle de fluxo foi descrita anteriormente e, a seguir, nós irá apenas descrever a janela de congestionamento.

O controle de congestionamento moderno foi adicionado ao TCP em grande parte por meio dos esforços de

Van Jacobson (1988). É uma história fascinante. A partir de 1986, a crescente população a clareza do início da Internet levou à primeira ocorrência do que ficou conhecido como **colapso da congestão**, um período prolongado durante o qual goodput caiu precipitadamente significativamente (ou seja, por mais de um fator de 100) devido ao congestionamento na rede. Jacob-

filho (e muitos outros) começou a entender o que estava acontecendo e remediar a situação.

A correção de alto nível que Jacobson implementou foi aproximar um AIMD janela de congestionamento. A parte interessante e grande parte da complexidade do TCP com controle de gestão, é como ele adicionou isso a uma implementação existente sem mudanças em qualquer um dos formatos de mensagem, o que o tornou imediatamente implantável. Para começar, ele

observaram que a perda de pacotes é um sinal adequado de congestionamento. Este sinal vem a um pouco tarde (pois a rede já está congestionada) mas é bastante confiável. Depois de De qualquer forma, é difícil construir um roteador que não descarte pacotes quando está sobrecarregado.

É improvável que esse fato mude. Mesmo quando as memórias de terabyte parecem armazenar em buffer

um grande número de pacotes, provavelmente teremos redes de terabit / s para preencher essas memórias.

No entanto, usar a perda de pacotes como um sinal de congestionamento depende da transmissão rara sendo relativamente rara. Normalmente, não é o caso de links sem fio, como

802.11, razão pela qual eles incluem seu próprio mecanismo de retransmissão no link

camada. Por causa das retransmissões sem fio, perda de pacotes da camada de rede devido a erros de transmissão são normalmente mascarados em redes sem fio. Também é raro em outros links porque os fios e as fibras ópticas normalmente têm baixas taxas de erro de bit.

Todos os algoritmos TCP da Internet assumem que os pacotes perdidos são causados por controlar e monitorar tempos limite e procurar sinais de problemas da maneira como os mineiros observam

seus canários. Um bom cronômetro de retransmissão é necessário para detectar sinais de perda de pacotes

com precisão e em tempo hábil. Já discutimos como o TCP re-temporizador de transmissão inclui estimativas da média e variação na viagem de ida e volta

Trabalho de Jacobson. Dado um bom tempo limite de retransmissão, o remetente TCP pode rastrear o número pendente de bytes, que estão carregando a rede. Simplesmente parece na diferença entre os números de sequência que são transmitidos e saliente.

Agora parece que nossa tarefa é fácil. Tudo o que precisamos fazer é rastrear os congestionamentos janela de configuração, usando números de sequência e confirmação, e ajustar a janela de gestão usando uma regra AIMD. Como você poderia esperar, é mais complicado do que isso. Uma primeira consideração é que a forma como os pacotes são enviados para

a rede, mesmo em curtos períodos de tempo, deve ser combinada com a rede caminho. Caso contrário, o tráfego causará congestionamento. Por exemplo, considere um host com uma janela de congestionamento de 64 KB conectada a uma Ethernet comutada de 1 Gbps. E se o host envia a janela inteira de uma vez, essa explosão de tráfego pode viajar por um link ADSL lento de 1 Mbps mais adiante no caminho. A explosão que levou apenas meio milissegundo na linha de 1 Gbps irá obstruir a linha de 1 Mbps por meio segundo, interrompendo completamente protocolos como voz sobre IP. Este comportamento pode ser uma ideia de um protocolo projetado para causar congestionamento, mas não de um protocolo para controlar isto.

No entanto, descobrimos que podemos usar pequenas rajadas de pacotes para nossa vantagem. A Fig. 6-43 mostra o que acontece quando um remetente em uma rede rápida (o 1-Gbps link) envia uma pequena rajada de quatro pacotes para um receptor em uma rede lenta (o 1-Link Mbps) que é o gargalo ou a parte mais lenta do caminho. Inicialmente os quatro pacotes trafegam pelo link tão rápido quanto podem ser enviados pelo remetente. No roteador, eles são enfileirados enquanto são enviados porque leva mais tempo para enviar um pacote pelo link lento do que para receber o próximo pacote pelo link rápido. Mas a fila não é grande porque apenas um pequeno número de pacotes foi enviado de uma vez. Note o aumento do comprimento dos pacotes no link lento. O mesmo pacote, de 1 KB, digamos, é agora mais porque leva mais tempo para enviá-lo em um link lento do que em um rápido.

Link rápido
Link lento
(gargalo)
1: estouro de pacotes
enviado em link rápido
2: Filas de ruptura no roteador
e drena para o link lento
3: Receber pacotes acks
na taxa de link lento
4: Ack preservação lenta
tempo do link no remetente
Ack clock
Receptor
Remetente
...
...
...
...
...

Figura 6-43. Uma explosão de pacotes de um remetente e o relógio de confirmação de retorno. Eventualmente, os pacotes chegam ao receptor, onde são reconhecidos. Os tempos para as confirmações refletem os tempos em que os pacotes chegaram ao receptor após cruzar o link lento. Eles estão espalhados em comparação com os pacotes originais no link rápido. À medida que esses reconhecimentos viajam pela rede trabalham e voltam para o remetente, eles preservam esse tempo.

não enfileire e congestione nenhum roteador ao longo do caminho. Este tempo é conhecido como um **relógio de confirmação**. É uma parte essencial do TCP. Usando um relógio de confirmação, o TCP suaviza

tráfego para fora e evita filas desnecessárias nos roteadores.

Uma segunda consideração é que a regra AIMD levará muito tempo para alcançar um bom ponto de operação em redes rápidas se a janela de congestionamento for iniciada de um tamanho pequeno. Considere um caminho de rede modesto que pode suportar 10 Mbps com um RTT de 100 mseg. A janela de congestionamento apropriada é o atraso da largura de banda produto, que é de 1 Mbit ou 100 pacotes de 1250 bytes cada. Se o congestionamento vencer dow começa em 1 pacote e aumenta em 1 pacote a cada RTT, serão 100 RTTs ou 10 segundos antes de a conexão funcionar na taxa certa. Aquilo é um

muito tempo para esperar apenas para obter a velocidade certa para uma transferência. Nós poderíamos reduzir isso

tempo de inicialização começando com uma janela inicial maior, digamos de 50 pacotes. Mas isso a janela seria muito grande para links lentos ou curtos. Isso causaria congestionamento se usado de uma vez, como acabamos de descrever.

Em vez disso, a solução que Jacobson escolheu para lidar com essas duas considerações é uma mistura de aumento linear e multiplicativo. Quando uma conexão é estabelecida, o remetente inicializa a janela de congestionamento para um pequeno valor inicial de no máximo quatro

segmentos; os detalhes são descritos no RFC 3390, e o uso de quatro segmentos é um aumento de um valor inicial anterior de um segmento com base na experiência. o o remetente então envia a janela inicial. Os pacotes levarão um tempo de ida e volta para ser reconhecido. Para cada segmento que é reconhecido antes da retransmissão o temporizador de sessão dispara, o remetente adiciona o valor de um segmento de bytes para o conges-

janela de instalação. Além disso, como esse segmento foi reconhecido, agora há um a menos segmento na rede. O resultado é que cada segmento reconhecido permite mais dois segmentos a serem enviados. A janela de congestionamento está dobrando a cada rodada-hora de viagem.

Este algoritmo é chamado de **início lento**, mas não é lento, é exponencial crescimento - exceto em comparação com o algoritmo anterior que permitia um fluxo inteiro janela de controle ser enviada de uma só vez. A partida lenta é mostrada na Fig. 6-44. Em primeiro tempo de ida e volta, o remetente injeta um pacote na rede (e o receptor recebe um pacote). Dois pacotes são enviados no próximo tempo de ida e volta, depois quatro pacotes no terceiro tempo de ida e volta.

A inicialização lenta funciona bem em uma variedade de velocidades de link e tempos de ida e volta, e

usa um relógio de confirmação para corresponder à taxa de transmissões do remetente para o caminho da rede.

Dê uma olhada na forma como as confirmações retornam do remetente para o destinatário na Fig. 6-44. Quando o remetente obtém uma confirmação, aumenta o congestionamento janela de conexão por um e imediatamente envia dois pacotes para a rede. (1 pacote é o aumento de um; o outro pacote é uma substituição para o pacote que foi reconhecido e saiu da rede. Em todos os momentos, o número de

1 RTT, 4 pacotes
(o tubo está cheio)
Dados
Reconhecimento
Remetente TCP
Receptor TCP

Figura 6-44. Partida lenta a partir de uma janela de congestionamento inicial de um segmento. pacotes não confirmados são fornecidos pela janela de congestionamento.) No entanto, esses dois os pacotes não chegarão necessariamente ao receptor tão espaçados como quando foram enviados. Por exemplo, suponha que o remetente esteja em uma Ethernet de 100 Mbps. Cada pacote de 1250 bytes leva 100 µs para ser enviado. Portanto, o atraso entre os pacotes pode ser tão pequeno quanto 100 µseg. A situação muda se esses pacotes passarem por um 1- Link ADSL de Mbps em qualquer lugar ao longo do caminho. Agora leva 10 ms para enviar o mesmo pacote. Isso significa que o espaçamento mínimo entre os dois pacotes tem cresceu por um fator de 100. A menos que os pacotes tenham que esperar juntos em uma fila em um link posterior, o espaçamento permanecerá grande.

Na Fig. 6-44, este efeito é mostrado pela aplicação de um espaçamento mínimo entre pacotes de dados que chegam ao receptor. O mesmo espaçamento é mantido quando o receptor envia confirmações e, portanto, quando o remetente recebe a confirmação mentos. Se o caminho da rede for lento, as confirmações virão lentamente (após um atraso de um RTT). Se o caminho da rede for rápido, as confirmações virão rapidamente (novamente, após o RTT). Tudo o que o remetente precisa fazer é seguir o tempo do ack clock conforme ele injeta novos pacotes, que é o que a inicialização lenta faz.

Porque o início lento causa crescimento exponencial, eventualmente (e mais cedo, em vez mais tarde) ele enviará muitos pacotes para a rede muito rapidamente. Quando isso acontecer, as filas irão se acumular na rede. Quando as filas estão cheias, uma ou mais pacotes serão perdidos. Depois que isso acontecer, o emissor TCP atingirá o tempo limite quando uma confirmação não chega a tempo. Há evidências de crescimento lento de início muito rápido na Fig. 6-44. Após três RTTs, quatro pacotes estão na rede. Estes quatro pacotes levam um RTT inteiro para chegar ao receptor. Ou seja, um congestionamento janela de quatro pacotes é o tamanho certo para esta conexão. No entanto, como estes os pacotes são reconhecidos, o início lento continua a aumentar a janela de congestionamento, atingindo oito pacotes em outro RTT. Apenas quatro desses pacotes podem alcançar o receptor em um RTT, não importa quantos sejam enviados. Ou seja, o tubo de rede é cheio. Pacotes adicionais colocados na rede pelo remetente serão acumulados em

Página 600

576

A CAMADA DE TRANSPORTE

INDIVÍDUO. 6

filas do roteador, uma vez que não podem ser entregues ao receptor com rapidez suficiente. Vigarista-

a gestação e a perda de pacotes ocorrerão em breve.

Para manter o início lento sob controle, o remetente mantém um limite para a conexão íon chamado de **limite de início lento**. Inicialmente, esse valor é definido arbitrariamente alto, para o tamanho da janela de controle de fluxo, de modo que não limite a conexão. TCP continua aumentando a janela de congestionamento no início lento até que ocorra um tempo limite ou o

a janela de congestionamento excede o limite (ou a janela do receptor está cheia).

Sempre que uma perda de pacote é detectada, por exemplo, por um tempo limite, o início lento o limite é definido como metade da janela de congestionamento e todo o processo é reiniciado. A ideia é que a janela atual é muito grande porque causou con gestion anteriormente que só agora é detectado por um tempo limite. Metade da janela, que foi usado com sucesso em um momento anterior, é provavelmente uma estimativa melhor para um

janela de congestionamento que está próxima da capacidade do caminho, mas não causará perda. No nosso exemplo na Figura 6.44, aumentar a janela de congestionamento para oito pacotes pode

causar perda, enquanto a janela de congestionamento de quatro pacotes no RTT anterior foi o valor certo. A janela de congestionamento é então redefinida para seu pequeno valor inicial e retoma o início lento.

Sempre que o limite de início lento é ultrapassado, o TCP muda de início lento para aumento aditivo. Neste modo, a janela de congestionamento é aumentada em um segmento a cada tempo de ida e volta. Como o início lento, isso geralmente é implementado com um aumento para cada segmento que é reconhecido, em vez de um aumento uma vez por RTT. Chame a janela de congestionamento $cwnd$ e o tamanho máximo do segmento MSS . UMA aproximação comum é aumentar $cwnd$ em $(MSS \times MSS) / cwnd$ para cada um dos Pacotes $cwnd / MSS$ que podem ser confirmados. Este aumento não precisa ser velozes. A ideia é que uma conexão TCP gaste muito tempo com seus con janela de gestação perto do valor ideal - não tão pequena que a taxa de transferência será baixo, e não tão grande que possa ocorrer congestionamento.

O aumento aditivo é mostrado na Fig. 6-45 para a mesma situação do início lento. Em ao final de cada RTT, a janela de congestionamento do remetente cresceu o suficiente para pode injetar um pacote adicional na rede. Comparado ao início lento, o a taxa linear de crescimento é muito mais lenta. Faz pouca diferença para pequenos congesjanelas de ção, como é o caso aqui, mas uma grande diferença no tempo que leva para crescer a janela de congestionamento para 100 segmentos, por exemplo.

Há algo mais que podemos fazer para melhorar o desempenho também. o defeito no esquema até agora está aguardando um tempo limite. Timeouts são relativamente longos porque eles devem ser conservadores. Depois que um pacote é perdido, o receptor não pode confirmar após isso, então o número de confirmação permanecerá fixo, e o remetente não será capaz de enviar novos pacotes para a rede porque a janela de gestação permanece cheia. Esta condição pode continuar por um período relativamente longo período até o temporizador disparar e o pacote perdido ser retransmitido. Nesse estágio, o TCP lento começa novamente.

Existe uma maneira rápida para o remetente reconhecer que um de seus pacotes tem foi perdido. Conforme os pacotes além do pacote perdido chegam ao receptor, eles acionam

Página 601

SEC. 6,5
OS PROTOCOLOS DE TRANSPORTE DA INTERNET: TCP

577

$cwnd = 2$
1 RTT, 2 pacotes
 $cwnd = 3$
 $cwnd = 4$
 $cwnd = 5$
1 RTT, 3 pacotes
1 RTT, 4 pacotes
1 RTT, 4 pacotes
(o tubo está cheio)
Dados
Reconhecimento
Remetente TCP
Receptor TCP
 $cwnd = 1$
1 RTT, 1 pacote

Figura 6-45. Aumento aditivo de uma janela de congestionamento inicial de um segmento. agradecimentos que retornam ao remetente. Esses reconhecimentos trazem o mesmo número de confirmação. Eles são chamados de **confirmações duplicadas**. Cada vez que o remetente recebe uma confirmação duplicada, é provável que um outro pacote chegou ao receptor e o pacote perdido ainda não apareceu. Como os pacotes podem seguir caminhos diferentes através da rede, eles podem chegar fora de serviço. Isso irá disparar confirmações duplicadas, mesmo que nenhum pacote ets foram perdidos. No entanto, isso é incomum na Internet na maior parte do tempo. Quando há reordenamento em vários caminhos, os pacotes recebidos são geralmente não reordenado muito. Assim, o TCP assume um tanto arbitrariamente que três duplicatas os reconhecimentos de categoria implicam que um pacote foi perdido. A identidade do perdido o pacote também pode ser inferido do número de confirmação. É o próprio

próximo pacote na sequência. Este pacote pode então ser retransmitido imediatamente, antes o tempo limite de retransmissão é acionado.

Essa heurística é chamada de **retransmissão rápida**. Depois de disparar, o início lento O limite ainda está definido para metade da janela de congestionamento atual, assim como com um tempo limite.

O início lento pode ser reiniciado definindo a janela de congestionamento para um pacote. Com neste tamanho de janela, um novo pacote será enviado após o tempo de ida e volta que ele leva para reconhecer o pacote retransmitido junto com todos os dados que foram enviado antes que a perda fosse detectada.

Uma ilustração do algoritmo de congestionamento que construímos até agora é mostrada em Fig. 6-46. Esta versão do TCP é chamada TCP Tahoe após o 4.2BSD Tahoe re arrendamento em 1988 em que foi incluído. O tamanho máximo do segmento aqui é 1 KB. Inicialmente, a janela de congestionamento era de 64 KB, mas ocorreu um tempo limite, então o espera é definida para 32 KB e a janela de congestionamento para 1 KB para transmissão 0. O a janela de congestionamento cresce exponencialmente até atingir o limite (32 KB). o

Página 602

578

A CAMADA DE TRANSPORTE
INDIVÍDUO. 6

janela é aumentada cada vez que chega uma nova confirmação, em vez de continuamente, o que leva ao padrão de escada discreto. Depois que o limite é passado sed, a janela cresce linearmente. É aumentado em um segmento a cada RTT.

5
Rodada de transmissão (RTTs)
Aditivo
aumentar
Limiar 32 KB
Pacote
perda
Congestionamento
janela
(KB
ou
pacotes)
10
15
20
30
35
40
25
2
4
6
8
10
12
14
16
18
20
22
24
Início lento
0
Limiar 20 KB

Figura 6-46. Início lento seguido por aumento aditivo no TCP Tahoe.

As transmissões na rodada 13 são azaradas (eles deveriam saber), e um deles está perdido na rede. Isso é detectado quando três confirmações duplicadas chegam. Nesse momento, o pacote perdido é retransmitido, o limite é definido como metade da janela atual (agora 40 KB, então a metade é 20 KB), e o início lento é iniciado tudo de novo. Reiniciar com uma janela de congestionamento de um pacote leva um tempo de ida e volta para que todos os dados transmitidos anteriormente saiam da rede e ser confirmado, incluindo o pacote retransmitido. A janela de congestionamento cresce com início lento como antes, até atingir o novo limite de 20 KB. Nesse momento, o crescimento torna-se linear novamente. Vai continuar desta forma

até que outra perda de pacote seja detectada por meio de confirmações duplicadas ou um tempo limite

(ou a janela do receptor se torna o limite).

TCP Tahoe (que incluía bons temporizadores de retransmissão) forneceu um funcionamento algoritmo de controle de congestionamento que resolveu o problema do colapso do congestionamento.

Jacobson percebeu que é possível fazer ainda melhor. No momento da rápida retransmissão, a conexão está funcionando com uma janela de congestionamento que é muito grande, mas ainda está funcionando com um relógio de confirmação funcionando. Cada vez que outra dupli-

Se o reconhecimento chegar, é provável que outro pacote tenha deixado a rede.

Usar confirmações duplicadas para contar os pacotes na rede, torna possível deixar alguns pacotes sair da rede e continuar a enviar um novo pacote para cada confirmação duplicada adicional.

A **recuperação rápida** é a heurística que implementa esse comportamento. É temporário modo que visa manter o relógio de confirmação funcionando com uma janela de congestionamento que

é o novo limite, ou metade do valor da janela de congestionamento no momento do

Página 603

SEC. 6,5

OS PROTOCOLOS DE TRANSPORTE DA INTERNET: TCP

579

retransmissão rápida. Para fazer isso, as confirmações duplicadas são contadas (incluindo os três que dispararam a retransmissão rápida) até o número de pacotes no rede caiu para o novo limite. Isso leva cerca de metade do tempo de ida e volta.

A partir de então, um novo pacote pode ser enviado para cada confirmação duplicada que é recebido. Um tempo de ida e volta após a retransmissão rápida, o pacote perdido foram reconhecidos. Naquela época, o fluxo de confirmações duplicadas será interrompido e o modo de recuperação rápida será encerrado. A janela de congestionamento será

definido para o novo limite de início lento e cresce por aumento linear.

O resultado dessa heurística é que o TCP evita o início lento, exceto quando o a conexão é iniciada pela primeira vez e quando ocorre um tempo limite. O último ainda pode acontecer

quando mais de um pacote é perdido e a retransmissão rápida não recupera ade-quately. Em vez de inícios lentos repetidos, a janela de congestionamento de um congestionamento conexão segue um padrão **dente de serra** de aumento aditivo (por um segmento a cada RTT) e diminuição multiplicativa (pela metade em um RTT). Este é exatamente o AIMD regra que procuramos implementar.

Esse comportamento de dente de serra é mostrado na Figura 6-47. É produzido pela TCP Reno, nomeado após o lançamento do 4.3BSD Reno em 1990 no qual foi incluído. TCP Reno é essencialmente TCP Tahoe mais recuperação rápida. Depois de um início lento, o a janela de congestionamento sobe linearmente até que uma perda de pacote seja detectada por ac-conhecimentos. O pacote perdido é retransmitido e a recuperação rápida é usada para mantenha o relógio de confirmação funcionando até que a retransmissão seja confirmada. Naquela hora,

a janela de congestionamento é retomada a partir do novo limite de início lento, em vez de de 1. Este comportamento continua indefinidamente, e a conexão gasta a maior parte do o tempo com sua janela de congestionamento perto do valor ótimo da banda produto de atraso de largura.

5

Rodada de transmissão (RTTs)

Aditivo

aumentar

Pacote

perda

Congestionamento

janela

(KB)

ou
pacotes)
10
15
20
30
35
40
25
4
8
12
16
20
24
28
32
36
40
44
48
Início lento
0
Thresh.
Limite
Rápido
recuperação
Multiplicativo
diminuir
Limite

Figura 6-47. Recuperação rápida e o padrão dente de serra do TCP Reno.

TCP Reno com seus mecanismos para ajustar a janela de congestionamento tem formou a base para o controle de congestionamento do TCP por mais de duas décadas. O máximo de

Página 604

580

A CAMADA DE TRANSPORTE INDIVÍDUO. 6

as mudanças nos anos intermediários ajustaram esses mecanismos em menor formas, por exemplo, alterando as opções da janela inicial e removendo várias ambigüidades. Algumas melhorias foram feitas para a recuperação de duas ou mais perdas em uma janela de pacotes. Por exemplo, o TCP NewReno versão usa um avanço parcial do número de confirmação após uma retransmissão para encontrar e reparar outra perda (Hoe, 1996), conforme descrito no RFC 3782. Desde os meados da década de 1990, surgiram várias variações que seguem os princípios que definimos escritas, mas usam leis de controle ligeiramente diferentes. Por exemplo, o Linux usa uma variante chamado CUBIC TCP (Ha et al., 2008) e o Windows inclui uma variante chamada Combra TCP (Tan et al., 2006).

Duas mudanças maiores também afetaram as implementações do TCP. Primeiro, muito de a complexidade do TCP vem de inferir de um fluxo de confirmação duplicada edgegments quais pacotes chegaram e quais pacotes foram perdidos. o o número de confirmação cumulativo não fornece essa informação. Um simples fix é o uso de **SACK** (reconhecimentos seletivos), que lista até três intervalos de bytes que foram recebidos. Com essas informações, o remetente pode decidir mais diretamente quais pacotes retransmitir e rastrear os pacotes em vôo para implementar a janela de congestionamento.

Quando o remetente e o destinatário estabelecem uma conexão, cada um deles envia o *SACK* opção TCP permitida para sinalizar que eles entendem confirmações seletivas.

Depois que o SACK é habilitado para uma conexão, ele funciona conforme mostrado na Figura 6-48. Estão-

ceiver usa o campo de *número de confirmação* TCP da maneira normal, como um reconhecimento cumulativo do byte mais alto na ordem que foi recebido.

Quando recebe o pacote 3 fora de serviço (porque o pacote 2 foi perdido), ele envia um Opção SACK para os dados recebidos junto com a confirmação cumulativa (duplicada) edgegment para o pacote 1. A opção SACK dá os intervalos de bytes que foram ultrapassado o número dado pela confirmação cumulativa. O primeiro intervalo é o pacote que acionou a confirmação duplicada. Nas próximas

intervalos, se presentes, são blocos mais antigos. Até três intervalos são comumente usados. Por o pacote de tempo 6 é recebido, dois intervalos de bytes SACK são usados para indicar que pacote 6 e pacotes 3 a 4 foram recebidos, além de todos os pacotes até pacote 1. A partir das informações em cada *opção de SACK* que recebe, o remetente pode decidir quais pacotes retransmitir. Neste caso, a retransmissão dos pacotes 2 e 5 seria uma boa ideia.

SACK é uma informação estritamente consultiva. A detecção real de perda usando dup- autorizar reconhecimentos e ajustes para a janela de congestionamento proceda apenas como antes. No entanto, com SACK, o TCP pode se recuperar mais facilmente de situações em em que vários pacotes são perdidos quase ao mesmo tempo, uma vez que o remetente TCP sabe quais pacotes não foram recebidos. O SACK agora está amplamente implantado. isto é descrito na RFC 2883, e o controle de congestionamento TCP usando SACK é descrito no RFC 3517.

A segunda mudança é o uso de ECN (Notificação Explícita de Congestionamento) em além da perda de pacotes como um sinal de congestionamento. ECN é um mecanismo de camada IP para

Página 605

SEC. 6,5
OS PROTOCOLOS DE TRANSPORTE DA INTERNET: TCP

581

6
5
4
3
2
1
Pacotes perdidos
ACK: 1
ACK: 1
SACK: 3
ACK: 1
SACK: 3-4
ACK: 1
SACK: 6, 3-4
Remetente
Receptor
Retransmitir 2 e 5!

Figura 6-48. Agradecimentos seletivos.

notificar hosts de congestionamento que descrevemos na Seç. 5.3.4. Com ele, o TCP re- ceiver pode receber sinais de congestionamento do IP.

O uso de ECN é habilitado para uma conexão TCP quando o remetente e o re- ceiver indicam que eles são capazes de usar ECN, definindo o *ECE* e *CWR* bits durante o estabelecimento da conexão. Se ECN for usado, cada pacote que carrega um O segmento TCP é sinalizado no cabeçalho IP para mostrar que pode transportar um sinal ECN. Os roteadores que suportam ECN definirão um sinal de congestionamento nos pacotes que podem transportar

ECN sinaliza quando o congestionamento está se aproximando, em vez de descartar esses pacotes após o congestionamento.

O receptor TCP é informado se algum pacote que chega transporta um ECN con sinal de gestação. O receptor então usa o sinalizador *ECE* (ECN Echo) para sinalizar o TCP remetente que seus pacotes sofreram congestionamento. O remetente diz ao receptor que ouviu o sinal usando o sinalizador *CWR* (janela reduzida de congestionamento).

O remetente do TCP reage a essas notificações de congestionamento exatamente da mesma maneira como acontece com a perda de pacotes detectada por meio de confirmações duplicadas. No entanto, a situação é estritamente melhor. O congestionamento foi detectado e não pacote foi prejudicado de alguma forma. ECN é descrito em RFC 3168. Requer ambos suporte a host e roteador e ainda não é amplamente utilizado na Internet.

Para obter mais informações sobre o conjunto completo de comportamentos de controle de congestionamento que são implementados em TCP, consulte RFC 5681.

6.5.11 O Futuro do TCP

Como o carro-chefe da Internet, o TCP tem sido usado para muitas aplicações

e estendido ao longo do tempo para fornecer bom desempenho em uma ampla gama de redes. Muitas versões são implantadas com implementações ligeiramente diferentes do que a classe algoritmos sic que descrevemos, especialmente para controle de congestionamento e robustez contra ataques. É provável que o TCP continue a evoluir com a Internet. Nós mencionará duas questões específicas.

O primeiro é que o TCP não fornece a semântica de transporte que todos os aplicativos deseja. Por exemplo, alguns aplicativos desejam enviar mensagens ou registros cujos limites precisam ser preservados. Outros aplicativos funcionam com um grupo de

Página 606

582

A CAMADA DE TRANSPORTE

INDIVÍDUO. 6

conversas relacionadas, como um navegador da Web que transfere vários objetos de o mesmo servidor. Ainda outros aplicativos querem melhor controle sobre os caminhos de rede que eles usam. O TCP com sua interface de soquetes padrão não atende a essas necessidades bem. Essencialmente, o aplicativo tem o fardo de lidar com qualquer problema que não seja resolvido por TCP. Isso levou a propostas de novos protocolos que forneciam um interface ligeiramente diferente. Dois exemplos são SCTP (Stream Control Transmission Protocol), definido na RFC 4960, e SST (Structured Stream Transport) (Ford, 2007). No entanto, sempre que alguém propõe mudar algo que

funcionou tão bem por tanto tempo, sempre há uma grande batalha entre os " Usuários estão exigindo mais recursos " e " Se não está quebrado, não conserte ".

O segundo problema é o controle de congestionamento. Você pode ter esperado que este seja um problema resolvido após nossas deliberações e os mecanismos que foram desenvolvidos optou ao longo do tempo. Não tão. A forma de controle de congestionamento TCP que descrevemos, e que é amplamente utilizado, baseia-se na perda de pacotes como sinal de congestionamento.

Quando Padhye et al. (1998) modelou a taxa de transferência de TCP com base no padrão dente de serra

tern, eles descobriram que a taxa de perda de pacotes deve cair rapidamente com o aumento Rapidez. Para alcançar uma taxa de transferência de 1 Gbps com um tempo de ida e volta de 100 ms e 1500

pacotes de bytes, um pacote pode ser perdido aproximadamente a cada 10 minutos. Aquilo é um taxa de perda de pacotes de 2×10^{-8}

, que é incrivelmente pequeno. É muito raro para servir como um bom sinal de congestionamento e qualquer outra fonte de perda (por exemplo, transmissão de pacotes

taxas de erro de sessão de 10⁻⁷

) pode facilmente dominá-lo, limitando o rendimento.

Essa relação não foi um problema no passado, mas as redes estão conseguindo

cada vez mais rápido, levando muitas pessoas a revisitar o controle de congestionamento. Um possibil-

é usar um controle de congestionamento alternativo no qual o sinal não é perda de pacote em absoluto. Demos vários exemplos na Seç. 6,2 O sinal pode ser de ida e volta, que cresce quando a rede fica congestionada, como é usado pelo FAST TCP (Wei et al., 2006). Outras abordagens também são possíveis, e o tempo dirá qual é ao melhor.

6.6 QUESTÕES DE DESEMPENHO

Problemas de desempenho são muito importantes em redes de computadores. Quando centenas ou milhares de computadores estão interconectados, interações complexas, com consequências vistas, são comuns. Freqüentemente, essa complexidade leva a um baixo desempenho força e ninguém sabe por quê. Nas seções a seguir, examinaremos muitos problemas relacionados ao desempenho da rede para ver quais tipos de problemas existem e o que pode ser feito sobre eles.

Infelizmente, entender o desempenho da rede é mais uma arte do que uma ciência

ence. Há pouca teoria subjacente que seja realmente útil na prática. o melhor que podemos fazer é fornecer algumas regras práticas obtidas com a dura experiência e pressão
exemplos retirados do mundo real. Adiamos essa discussão até que estudou a camada de transporte porque o desempenho que os aplicativos recebem

Página 607

SEC. 6,6
PROBLEMAS DE DESEMPENHO

583

depende do desempenho combinado das camadas de transporte, rede e link, e ser capaz de usar o TCP como exemplo em vários lugares.

Nas próximas seções, veremos seis aspectos do desempenho da rede:

1. Problemas de desempenho.
2. Medir o desempenho da rede.
3. Projeto de host para redes rápidas.
4. Processamento rápido de segmentos.
5. Compressão de cabeçalho.
6. Protocolos para redes " long fat ".

Esses aspectos consideram o desempenho da rede tanto no host quanto na rede trabalho e à medida que as redes aumentam em velocidade e tamanho.

6.6.1 Problemas de desempenho em redes de computadores

Alguns problemas de desempenho, como congestionamento, são causados por sobrecargas da fonte. Se mais tráfego chegar repentinamente a um roteador do que o roteador pode lidar com o problema, o congestionamento aumentará e o desempenho será prejudicado. Nós estudamos congestionamento detalhes neste capítulo e no anterior.

O desempenho também se degrada quando há um desequilíbrio estrutural de recursos. Para por exemplo, se uma linha de comunicação gigabit estiver conectada a um PC low-end, o pobre host não será capaz de processar os pacotes recebidos rápido o suficiente e alguns estarão perdidos. Esses pacotes serão eventualmente retransmitidos, adicionando atraso, desperdiçando largura de banda e geralmente reduzindo o desempenho.

Sobrecargas também podem ser disparadas de forma síncrona. Por exemplo, se um segmento contém um parâmetro inválido (por exemplo, a porta para a qual é destinado), em muitos casos o receptor enviará de volta uma notificação de erro. Agora considere o que pode acontecer se um segmento inválido for transmitido para 1000 máquinas: cada uma pode enviar de volta uma mensagem de erro. A **tempestade de transmissão** resultante pode paralisar a rede. UDP sofreu com este problema até que o protocolo ICMP foi alterado para fazer com que os hosts se abstêm de responder a erros nos segmentos UDP enviados para endereços de elenco. As redes sem fio devem ser particularmente cuidadosas para evitar desmarcadas

respostas de transmissão porque a transmissão ocorre naturalmente e a banda sem fio largura é limitada.

Um segundo exemplo de sobrecarga síncrona é o que acontece após uma falha de energia. Quando a energia volta, todas as máquinas simultaneamente iniciam a reinicialização. Uma sequência de reinicialização típica pode exigir primeiro ir a algum Servidor (DHCP) para descobrir a verdadeira identidade de alguém e, em seguida, para algum servidor de arquivos para obter uma cópia do sistema operacional. Se centenas de máquinas em um data center fizerem isso de uma vez, o servidor provavelmente entrará em colapso sob a carga.

Página 608

584

A CAMADA DE TRANSPORTE
INDIVÍDUO. 6

Mesmo na ausência de sobrecargas síncronas e na presença de suficientes

recursos, pode ocorrer baixo desempenho devido à falta de ajuste do sistema. Para exemplo, se uma máquina tiver bastante poder de CPU e memória, mas não o suficiente memória foi alocada para espaço de buffer, o controle de fluxo irá desacelerar o segmento recepção e limite de desempenho. Este era um problema para muitas conexões TCP à medida que a Internet se tornou mais rápida, mas o tamanho padrão da janela de controle de fluxo ficou fixo em 64 KB.

Outro problema de ajuste é definir tempos limite. Quando um segmento é enviado, um cronômetro é definido para proteger contra a perda do segmento. Se o tempo limite for definido muito curto, desnecessário

ocorrerão retransmissões sárias, obstruindo os fios. Se o tempo limite for definido muito longo, atrasos desnecessários ocorrerão após a perda de um segmento. Outros parâmetros ajustáveis incluem quanto tempo esperar pelos dados sobre os quais pegar carona antes de enviar um reconhecimento e quantas retransmissões fazer antes de desistir.

Outro problema de desempenho que ocorre com aplicativos em tempo real como o áudio e o vídeo estão instáveis. Ter largura de banda suficiente em média não é suficiente para um bom desempenho. Também são necessários pequenos atrasos na transmissão. Consistentemente

alcançar pequenos atrasos exige uma engenharia cuidadosa da carga na rede, suporte de qualidade de serviço nas camadas de link e rede, ou ambas.

6.6.2 Medição de Desempenho da Rede

Quando uma rede funciona mal, seus usuários costumam reclamar para as pessoas que executam, exigindo melhorias. Para melhorar o desempenho, os operadores devem primeiro determinar exatamente o que está acontecendo. Para descobrir o que realmente está acontecendo, os operadores devem fazer medições. Nesta seção, veremos a rede por medições de desempenho. Grande parte da discussão abaixo é baseada no seminal trabalho de Mogul (1993).

As medições podem ser feitas de diferentes maneiras e em muitos locais (tanto em a pilha de protocolo e fisicamente). O tipo mais básico de medição é começar um cronômetro ao iniciar alguma atividade e ver quanto tempo leva essa atividade. Para exemplo, saber quanto tempo leva para um segmento ser reconhecido é uma chave medição. Outras medições são feitas com contadores que registram a frequência algum evento aconteceu (por exemplo, número de segmentos perdidos). Finalmente, um é frequentemente

interessado em saber a quantidade de algo, como o número de bytes processados em um determinado intervalo de tempo.

Medir o desempenho e os parâmetros da rede tem muitas armadilhas potenciais. Listamos alguns deles aqui. Qualquer tentativa sistemática de medir a rede por o desempenho deve ter o cuidado de evitá-los.

Certifique-se de que o tamanho da amostra é grande o suficiente

Não meça o tempo para enviar um segmento, mas repita a medição, digamos, um milhão de vezes e calcule a média. Efeitos de inicialização, como o 802.16 NIC ou modem a cabo obtendo uma reserva de largura de banda após um período inativo, pode

desacelere o primeiro segmento e o enfileiramento introduzirá variabilidade. Ter um grande sample irá reduzir a incerteza na média medida e no desvio padrão. Isto a incerteza pode ser calculada usando fórmulas estatísticas padrão.

Certifique-se de que as amostras são representativas

Idealmente, toda a sequência de um milhão de medições deve ser repetida em diferentes momentos do dia e da semana para ver o efeito de diferentes redes condições da quantidade medida. Medições de congestionamento, por exemplo, são de pouca utilidade se forem feitos em um momento em que não há congestionamento. Algumas vezes os resultados podem ser contra-intuitivos no início, como forte congestionamento em 11

Um . H ., E um P . M ., mas sem congestionamento ao meio-dia (quando todos os usuários estão almoçando).

Com redes sem fio, a localização é uma variável importante por causa do sinal propagação. Mesmo um nó de medição colocado perto de um cliente sem fio pode não observar os mesmos pacotes que o cliente devido a diferenças nas antenas. É melhor para fazer medições do cliente sem fio em estudo para ver o que ele vê.

Caso contrário, é possível usar técnicas para combinar as medições sem fio tiradas em diferentes pontos de vista para obter uma imagem mais completa do que está acontecendo em (Mahajan et al., 2006).

O cache pode causar estragos nas medições

Repetir uma medição muitas vezes retornará uma resposta inesperadamente rápida se os protocolos usam mecanismos de cache. Por exemplo, buscar uma página da Web ou procurar um nome DNS (para encontrar o endereço IP) pode envolver uma troca de rede na primeira vez e, em seguida, retornar a resposta de um cache local sem enviar pacotes pela rede. Os resultados de tal medição são essencialmente inútil (a menos que você queira medir o desempenho do cache).

O armazenamento em buffer pode ter um efeito semelhante. Os testes de desempenho TCP / IP foram conhecidos por relatar que o UDP pode atingir um desempenho substancialmente superior ao que a rede permite. Como isso ocorre? Uma chamada para UDP normalmente retorna o controle como assim que a mensagem for aceita pelo kernel e adicionada à transmissão fila de espera. Se houver espaço de buffer suficiente, cronometrar 1000 chamadas UDP não significa que todos os dados foram enviados. A maioria deles ainda pode estar no kernel, mas o programa de teste de desempenho pensa que todos foram transmitidos.

Recomenda-se cuidado para ter certeza absoluta de que você entende como os dados podem ser armazenados em cache e em buffer como parte de uma operação de rede.

Certifique-se de que nada inesperado está acontecendo durante seus testes

Fazendo medições ao mesmo tempo que algum usuário decidiu executar um videoconferência em sua rede geralmente dará resultados diferentes do que se houvesse não é uma videoconferência. É melhor executar testes em uma rede ociosa e criar o

Página 610

586

A CAMADA DE TRANSPORTE
INDIVÍDUO. 6

toda a carga de trabalho sozinho. Mesmo essa abordagem tem armadilhas, no entanto. Enquanto você

poderia pensar que ninguém vai usar a rede em 3 A . M ., Pode ser quando o programa de backup automático começa a copiar todos os discos para a fita. Ou, pode haver Pode haver tráfego pesado para suas maravilhosas páginas da Web em fusos horários distantes. As redes sem fio são desafiadoras a esse respeito porque muitas vezes não são possível separá-los de todas as fontes de interferência. Mesmo se não houver outro redes sem fio enviando tráfego nas proximidades, alguém pode colocar pipoca no micro-ondas e causar inadvertidamente interferência que degrada o desempenho do 802.11. Por estes motivos filhos, é uma boa prática monitorar a atividade geral da rede para que você possa pelo menos perceba quando algo inesperado acontecer.

Tenha cuidado ao usar um relógio de granulação grossa

Os relógios do computador funcionam incrementando algum contador em intervalos regulares. Por exemplo, um cronômetro de milissegundos adiciona 1 a um contador a cada 1 ms. Usando tal temporizador para medir um evento que leva menos de 1 mseg é possível, mas requer algum Cuidado. Alguns computadores têm relógios mais precisos, é claro, mas sempre há eventos mais curtos para medir também. Observe que os relógios nem sempre são tão precisos quanto o

precisão com a qual o tempo é retornado quando são lidos.

Para medir o tempo para fazer uma conexão TCP, por exemplo, o relógio (digamos, em milissegundos) deve ser lido quando o código da camada de transporte é inserido e

novamente quando for encerrado. Se o verdadeiro tempo de configuração da conexão for 300 μs, a diferença

a diferença entre as duas leituras será 0 ou 1, ambas erradas. No entanto, se a medição é repetida um milhão de vezes e o total de todas as medições é adicionado e dividido por um milhão, o tempo médio será preciso para melhor de 1 μseg.

Tenha cuidado ao extrapolar os resultados

Suponha que você faça medições com cargas de rede simuladas em execução de 0 (inativo) a 0,4 (40% da capacidade). Por exemplo, o tempo de resposta para enviar um pacote de voz sobre IP em uma rede 802.11 pode ser conforme mostrado pelos dados pontos e uma linha sólida através deles na Fig. 6-49. Pode ser tentador extrapolar linearmente, conforme mostrado pela linha pontilhada. No entanto, muitos resultados de filas envolvem um

fator de $1 / (1 - \rho)$, onde ρ é a carga, então os valores verdadeiros podem se parecer mais com o linha tracejada, que sobe muito mais rápido do que linearmente quando a carga fica alta. que ou seja, cuidado com os efeitos de contenção que se tornam muito mais pronunciados com carga alta.

6.6.3 Design de host para redes rápidas

Medir e mexer podem melhorar o desempenho consideravelmente, mas podem não substituir um bom design em primeiro lugar. Uma rede mal projetada pode ser melhorada apenas até certo ponto. Além disso, precisa ser redesenhado do zero.

SEC. 6,6
PROBLEMAS DE DESEMPENHO

587

5
4
3
2
1
0
Resposta
Tempo
1.0
0
0,1
0,2
0,3
0,4
0,5
Carga
0,6
0,7
0,8
0,9

Figura 6-49. Resposta em função da carga.

Nesta seção, apresentaremos algumas regras básicas para implementação de software mentação de protocolos de rede em hosts. Surpreendentemente, a experiência mostra que este costuma ser um gargalo de desempenho em redes que de outra forma seriam rápidas, por dois motivos.

Primeiro, NICs (placas de interface de rede) e roteadores já foram projetados (com suporte de hardware) para rodar na "velocidade do fio." Isso significa que eles podem processar

pacotes tão rapidamente quanto eles podem chegar no link. Segundo, o desempenho relevante é aquele obtido pelos aplicativos. Não é a capacidade do link, mas a taxa de transferência e o atraso após o processamento de rede e transporte.

Reducir sobrecargas de software melhora o desempenho, aumentando o rendimento e diminuição do atraso. Também pode reduzir a energia gasta na rede, que é uma consideração importante para computadores móveis. A maioria dessas ideias são de conhecimento comum para designers de rede há anos. Eles foram os primeiros afirmado explicitamente por Mogul (1993); nosso tratamento segue amplamente o dele. Outro fonte relevante é Metcalfe (1993).

A velocidade do host é mais importante do que a velocidade da rede

A longa experiência tem mostrado que em quase todas as redes rápidas, sistema operacional e a sobrecarga do protocolo domina o tempo real na transmissão. Por exemplo, em teoria, o tempo mínimo de RPC em uma Ethernet de 1 Gbps é 1 µseg, correspondendo a um mini-pedido mum (512 bytes) seguido por uma resposta mínima (512 bytes). Na prática, superando a sobrecarga de software e obtendo o tempo RPC em qualquer lugar próximo é uma conquista substancial. Isso raramente acontece na prática.

588

A CAMADA DE TRANSPORTE

INDIVÍDUO. 6

Da mesma forma, o maior problema em rodar a 1 Gbps é frequentemente obter os bits do buffer do usuário para a rede rápido o suficiente e tendo o recebimento o host processa-os tão rápido quanto eles chegam. Se você dobrar o host (CPU e mem- velocidade), muitas vezes você pode chegar perto de dobrar a taxa de transferência. Dobrando o a capacidade da rede não tem efeito se o gargalo estiver nos hosts.

Reduza a contagem de pacotes para reduzir as despesas gerais

Cada segmento tem uma certa quantidade de overhead (por exemplo, o cabeçalho), bem como dados (por exemplo, a carga útil). A largura de banda é necessária para ambos os componentes. O processamento é também necessário para ambos os componentes (por exemplo, processamento de cabeçalho e fazer a verificação

soma). Quando 1 milhão de bytes estão sendo enviados, o custo dos dados é o mesmo, não importa qual é o tamanho do segmento. No entanto, usar segmentos de 128 bytes significa 32 vezes como muito overhead por segmento com o uso de segmentos de 4 KB. A largura de banda e proc- despesas gerais aumentam rapidamente para reduzir o rendimento.

A sobrecarga por pacote nas camadas inferiores amplifica esse efeito. Cada chegada pacote causa uma nova interrupção se o host estiver acompanhando. Em um pipeline moderno processador, cada interrupção quebra o pipeline da CPU, interfere no cache, requer uma mudança no contexto de gerenciamento de memória, anula a previsão de ramificação tabela e força um número substancial de registros da CPU a serem salvos. Um n - vezes re- redução em segmentos enviados, portanto, reduz a interrupção e sobrecarga de pacote por um fator de n .

Você pode dizer que tanto as pessoas quanto os computadores são ruins em multitarefa. Isto observação está por trás do desejo de enviar pacotes MTU que são tão grandes quanto passarão ao longo do caminho da rede sem fragmentação. Mecanismos como o algoritmo de Nagle O ritmo e a solução de Clark também são tentativas de evitar o envio de pequenos pacotes.

Minimize o toque de dados

A maneira mais direta de implementar uma pilha de protocolos em camadas é com um módulo para cada camada. Infelizmente, isso leva à cópia (ou pelo menos ac- processar os dados em várias passagens) à medida que cada camada faz seu próprio trabalho. Para ex-

emplo, depois que um pacote é recebido pelo NIC, ele é normalmente copiado para um kernel buff- er. A partir daí, ele é copiado para um buffer da camada de rede para o processamento da camada de rede,

em seguida, para um buffer de camada de transporte para o processamento da camada de transporte e, finalmente, para o re-

processo de aplicação de ceiving. Não é incomum que um pacote de entrada seja copiado três ou quatro vezes antes de o segmento encerrado ser entregue.

Todas essas cópias podem degradar bastante o desempenho porque as operações de memória são uma ordem de magnitude mais lentas do que as instruções registrador-registrador. Por exemplo, se 20% das instruções realmente vão para a memória (ou seja, são falhas de cache), que é provavelmente ao tocar em pacotes de entrada, o tempo médio de execução da instrução é desacelerado por um fator de 2,8 ($0,8 \times 1 + 0,2 \times 10$). A assistência de hardware não ajuda aqui. O problema é o excesso de cópias pelo sistema operacional.

SEC. 6,6
PROBLEMAS DE DESEMPENHO

589

Um sistema operacional inteligente irá minimizar a cópia, combinando o processo de processamento de várias camadas. Por exemplo, TCP e IP são geralmente implementados juntos (como "TCP / IP") de modo que não seja necessário copiar a carga útil do pacote como o processamento muda da rede para a camada de transporte. Outro truque comum é executar várias operações dentro de uma camada em uma única passagem sobre os dados. Para exemplo, somas de verificação são freqüentemente calculadas durante a cópia dos dados (quando tem que ser

copiado) e a soma de verificação recém-calculada é anexada ao final.

Minimize as opções de contexto

Uma regra relacionada é que o contexto muda (por exemplo, do modo kernel para o modo de usuário)

são mortais. Eles têm as propriedades ruins de interrupção e cópia combinadas.

Esse custo é o motivo pelo qual os protocolos de transporte são frequentemente implementados no kernel. Gostar

reduzindo a contagem de pacotes, as mudanças de contexto podem ser reduzidas tendo a biblioteca pro-

procedimento que envia dados para buffer interno até que tenha uma quantidade substancial de eles. Da mesma forma, no lado receptor, pequenos segmentos de entrada devem ser coloridos selecionados juntos e passados para o usuário de uma só vez, em vez de individualmente, para minimizar as mudanças de contexto.

Na melhor das hipóteses, um pacote de entrada causa uma mudança de contexto do atual usuário para o kernel e, em seguida, uma mudança para o processo de recebimento para fornecer o novo

chegaram dados. Infelizmente, com alguns sistemas operacionais, contexto adicional mudanças acontecem. Por exemplo, se o gerenciador de rede é executado como um processo especial

no espaço do usuário, a chegada de um pacote provavelmente causará uma mudança de contexto do atual

usuário para o kernel, depois outro do kernel para o gerenciador de rede, a seguir baixado por outro de volta para o kernel e, finalmente, um do kernel para o re-processo de ceiving. Essa sequência é mostrada na Figura 6-50. Todas essas mudanças de contexto em cada pacote é uma perda de tempo de CPU e pode ter um efeito devastador na rede desempenho no trabalho.

Espaço do usuário

Espaço do kernel

1

2

3

4

Processo do usuário em execução no hora da chegada do pacote

Rede

Gerente

Recebendo

processo

Figura 6-50. Quatro opções de contexto para lidar com um pacote com uma rede de espaço do usuário gerente de trabalho.

590

A CAMADA DE TRANSPORTE
INDIVÍDUO. 6

Evitar o congestionamento é melhor do que se recuperar dele

A velha máxima de que um grama de prevenção vale um quilo de cura certamente retém para congestionamento da rede. Quando uma rede está congestionada, os pacotes são perdidos, a largura de banda é desperdiçada, atrasos inúteis são introduzidos e muito mais. Todos esses custos são desnecessários e a recuperação do congestionamento exige tempo e paciência. Não

que isso aconteça em primeiro lugar é melhor. Evitar o congestionamento é como obter sua vacinação DTP: dói um pouco na hora de tomar, mas previne alguns coisas que doeria muito mais no futuro.

Evite tempos limite

Timers são necessários em redes, mas devem ser usados com moderação e tempo-saídas devem ser minimizadas. Quando um cronômetro dispara, alguma ação é geralmente refeita turva. Se é realmente necessário repetir a ação, que assim seja, mas repetindo-a desnecessariamente é um desperdício.

A maneira de evitar trabalho extra é ter cuidado para que os temporizadores estejam um pouco ligados

o lado conservador. Um cronômetro que leva muito tempo para expirar adiciona uma pequena quantidade

de atraso extra para uma conexão no evento (improvável) de um segmento sendo perdido. UMA temporizador que dispara quando não deveria, consome recursos do host, desperdiça banda largura e coloca carga extra em talvez dezenas de roteadores sem um bom motivo.

6.6.4 Processamento de Segmento Rápido

Agora que cobrimos as regras gerais, veremos alguns métodos específicos para acelerar o processamento do segmento. Para obter mais informações, consulte Clark et al. (1989) e Chase et al. (2001).

A sobrecarga de processamento de segmento tem dois componentes: sobrecarga por segmento e sobrecarga por byte. Ambos devem ser atacados. A chave para o processamento rápido do segmento é

para separar o caso normal de sucesso (transferência de dados unilateral) e tratá-lo especialmente. Muitos protocolos tendem a enfatizar o que fazer quando algo vai errado (por exemplo, um pacote sendo perdido), mas para fazer com que os protocolos funcionem rapidamente, o designer

deve ter como objetivo minimizar o tempo de processamento quando tudo der certo. Minimizando o tempo de processamento quando ocorre um erro é secundário.

Embora uma sequência de segmentos especiais seja necessária para entrar no *ESTABLISHED*, uma vez lá, o processamento do segmento é direto até um lado começa a fechar a conexão. Vamos começar examinando o lado do envio no Estado *ESTABLISHED* quando há dados a serem transmitidos. Por uma questão de claridade, assumimos aqui que a entidade de transporte está no kernel, embora o mesmo as ideias se aplicam se for um processo de espaço do usuário ou uma biblioteca dentro do processo de envio. No

Figura 6-51, o processo de envio intercepta o kernel para fazer o SEND. A primeira coisa a entidade de transporte faz é testar para ver se este é o caso normal: o estado é *ESTABLISHED*, nenhum dos lados está tentando fechar a conexão, um regular (ou seja, não um

SEC. 6,6

PROBLEMAS DE DESEMPENHO

591

fora da banda) o segmento completo está sendo enviado e há espaço de janela suficiente disponível em

o receptor. Se todas as condições forem atendidas, nenhum outro teste será necessário e o caminho rápido

através da entidade de transporte de envio podem ser tomadas. Normalmente, esse caminho é seguido

a maior parte do tempo.

Trap no kernel para enviar segmento

Teste

Segmento passado para processo de recebimento

Teste

S

S

Enviando

processo

Processo de recebimento

Rede

Figura 6-51. O caminho rápido do remetente ao receptor é mostrado com uma linha grossa.

As etapas de processamento neste caminho estão sombreadas.

No caso usual, os cabeçalhos de segmentos de dados consecutivos são quase o mesmo. Para tirar proveito desse fato, um cabeçalho de protótipo é armazenado dentro do trans-entidade esportiva. No início do caminho rápido, ele é copiado o mais rápido possível para um buffer de rascunho, palavra por palavra. Esses campos que mudam de segmento para segmento são sobreescritos no buffer. Freqüentemente, esses campos são facilmente derivados do estado variáveis, como o próximo número de sequência. Um ponteiro para o cabeçalho completo do segmento

mais um ponteiro para os dados do usuário são então passados para a camada de rede. Aqui o a mesma estratégia pode ser seguida (não mostrada na Figura 6-51). Finalmente, a rede camada fornece o pacote resultante para a camada de enlace de dados para transmissão.

Como exemplo de como esse princípio funciona na prática, consideremos o TCP / IP.

A Fig. 6-52 (a) mostra o cabeçalho TCP. Os campos que são os mesmos entre cons- os segmentos ativos em um fluxo unilateral são sombreados. Toda a entidade de transporte de envio tem

a fazer é copiar as cinco palavras do cabeçalho do protótipo para o buffer de saída, preencher no próximo número de sequência (copiando-o de uma palavra na memória), calcule o soma de verificação e incrementa o número de sequência na memória. Ele pode então entregar o cabeçalho e dados para um procedimento IP especial para o envio de um segmento máximo regular ment. O IP então copia seu cabeçalho de protótipo de cinco palavras [veja a Figura 6-52 (b)] para o buffer, preenche o campo *Identificação* e calcula sua soma de verificação. O pacote é agora pronto para transmissão.

Agora, vamos examinar o processamento de caminho rápido no lado receptor da Figura 6.51. A etapa 1 é localizar o registro de conexão para o segmento de entrada. Para TCP, o

592

A CAMADA DE TRANSPORTE

INDIVÍDUO. 6

Número sequencial

(uma)

(b)

Soma de verificação do cabeçalho

Identificação

Porta de origem

Número de confirmação

Len não utilizado

Tamanho da janela

Checksum

Ponteiro urgente

Porto de destino

Deslocamento de fragmento

VER, IHL

Dif. Serv.

Comprimento total

TTL

Protocolo

Endereço de Origem

Endereço de destino

Dif. Serv.

Figura 6-52. (a) Cabeçalho TCP. (b) Cabeçalho IP. Em ambos os casos, eles são retirados de o protótipo sem alterações.

registro de conexão pode ser armazenado em uma tabela hash para a qual alguma função simples de os dois endereços IP e duas portas é a chave. Assim que o registro de conexão tiver localizado, os endereços e as portas devem ser comparados para verificar se o o registro correto foi encontrado.

Uma otimização que muitas vezes acelera ainda mais a pesquisa de registro de conexão é para manter um ponteiro para o último usado e tentar aquele primeiro. Clark et al. (1989) tentei isso e observei uma taxa de acerto superior a 90%.

O segmento é verificado para ver se é normal: o estado é *ESTABLISHED*, nenhum dos lados está tentando fechar a conexão, o segmento é completo, nenhum sinalizador especial é definido e o número de sequência é o esperado. Esses testes pegue apenas algumas instruções. Se todas as condições forem atendidas, um caminho rápido especial

O procedimento TCP é chamado.

O atalho atualiza o registro de conexão e copia os dados para o usuário.

Enquanto está copiando, ele também calcula a soma de verificação, eliminando uma passagem extra os dados. Se a soma de verificação estiver correta, o registro de conexão é atualizado e um conhecimento é enviado de volta. O esquema geral de primeiro fazer uma verificação rápida para veja se o cabeçalho é o esperado e, em seguida, ter um identificador de procedimento especial esse caso é chamado de **previsão de cabeçalho**. Muitas implementações de TCP o usam. Quando esta otimização e todas as outras discutidas neste capítulo são usadas juntas, é possível fazer com que o TCP rode a 90% da velocidade de uma memória para memória local cópia, presumindo que a própria rede seja rápida o suficiente.

Duas outras áreas onde grandes ganhos de desempenho são possíveis são o homem-tampão gerenciamento de agente e temporizador. O problema no gerenciamento de buffer é evitar cópia desnecessária, conforme mencionado acima. A gestão do temporizador é importante porque quase todos os temporizadores definidos não expiram. Eles são configurados para proteger contra o segmento

perda, mas a maioria dos segmentos e seus reconhecimentos chegam corretamente. Portanto, é importante otimizar o gerenciamento do cronômetro para o caso de cronômetros que raramente expiram.

Um esquema comum é usar uma lista vinculada de eventos de cronômetro classificados por expiração

Tempo. A entrada principal contém um contador que informa quantos carapatos faltam para expirar isto é. Cada entrada sucessiva contém um contador que informa quantos tiques após o

Página 617

SEC. 6,6

PROBLEMAS DE DESEMPENHO

593

entrada anterior. Assim, se os temporizadores expiram em 3, 10 e 12 ticks, respectivamente, os três contadores são 3, 7 e 2, respectivamente.

A cada tique do relógio, o contador na entrada principal é diminuído. Quando atinge zero, seu evento é processado e o próximo item da lista se torna o cabeçalho. Está contador não precisa ser alterado. Dessa forma, inserir e excluir cronômetros são operações caras, com tempos de execução proporcionais ao comprimento da lista.

Uma abordagem muito mais eficiente pode ser usada se o intervalo máximo do cronômetro for delimitado e conhecido com antecedência. Aqui, uma matriz chamada **roda de tempo** pode ser usado, conforme mostrado na Fig. 6-53. Cada slot corresponde a um tique do relógio. O atual o tempo mostrado é $T = 4$. Os temporizadores estão programados para expirar em 3, 10 e 12 ticks de agora. Se um novo cronômetro é definido para expirar repentinamente em sete ticks, uma entrada é feita

no slot 11. Da mesma forma, se o cronômetro definido para $T + 10$ tiver que ser cancelado, a lista começa

no slot 14 deve ser pesquisado e a entrada necessária removida. Observe que a matriz da Fig. 6-53 não pode acomodar temporizadores além de $T + 15$.

```
0  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
Slot  
0  
0  
0  
0
```

```
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
```

Ponteiro para lista de temporizadores para T + 12

Ponteiro para lista de temporizadores para T + 3

Ponteiro para lista de temporizadores para T + 10

Hora atual, T

Figura 6-53. Uma roda de cronometragem.

Quando o relógio marca, o ponteiro do tempo atual é avançado em um slot (circularly). Se a entrada agora apontada for diferente de zero, todos os seus temporizadores serão processados.

Muitas variações da ideia básica são discutidas por Varghese e Lauck (1987).

6.6.5 Compressão de cabeçalho

Há muito tempo que olhamos para as redes rápidas. Há mais por aí.

Vamos agora considerar o desempenho em redes sem fio e outras redes em que a banda largura é limitada. Reduzir a sobrecarga de software pode ajudar os computadores móveis a funcionar

Página 618

594

A CAMADA DE TRANSPORTE

INDIVÍDUO. 6

de forma mais eficiente, mas não faz nada para melhorar o desempenho quando a rede links são o gargalo.

Para usar bem a largura de banda, cabeçalhos de protocolo e cargas úteis devem ser transportados com

o mínimo de bits. Para cargas úteis, isso significa usar codificações compactas de informações informações, como imagens em formato JPEG em vez de bitmap ou documento formatos como PDF que incluem compactação. Também significa nível de aplicativo mecanismos de cache, como caches da Web, que reduzem as transferências em primeiro lugar. E quanto aos cabeçalhos de protocolo? Na camada de link, cabeçalhos para rede sem fio as obras são normalmente compactas porque foram projetadas com largura de banda escassa em mente. Por exemplo, cabeçalhos 802.16 têm identificadores de conexão curtos em vez de endereços mais longos. No entanto, protocolos de camada superior, como IP, TCP e UDP vêm em uma versão para todas as camadas de link e não são projetados com compactas cabeçalhos. Na verdade, o processamento simplificado para reduzir a sobrecarga de software geralmente leva

para cabeçalhos que não são tão compactos quanto poderiam ser (por exemplo, IPv6 tem um cabeçalhos mais fracamente compactados do que IPv4).

Os cabeçalhos de camada superior podem ser um impacto significativo no desempenho. Considere, por

exemplo, dados de voz sobre IP que estão sendo transportados com a combinação de IP, UDP e RTP. Esses protocolos requerem 40 bytes de cabeçalho (20 para IPv4, 8 para UDP e 12 para RTP). Com o IPv6 a situação é ainda pior: 60 bytes, incluindo o cabeçalho IPv6 de 40 bytes. Os cabeçalhos podem terminar como a maioria das trans-

dados perdidos e consomem mais da metade da largura de banda.

A **compressão do cabeçalho** é usada para reduzir a largura de banda ocupada pelos links por cabeçalhos de protocolo de camada superior. Esquemas especialmente projetados são usados em vez de

métodos de uso geral. Isso ocorre porque os cabeçalhos são curtos, então eles não comprimir bem individualmente, e a descompressão requer que todos os dados anteriores sejam refeitos

ceived. Este não será o caso se um pacote for perdido.

A compressão do cabeçalho obtém grandes ganhos usando o conhecimento do protocolo formato. Um dos primeiros esquemas foi projetado por Van Jacobson (1990) para

pressionando cabeçalhos TCP / IP em links seriais lentos. É capaz de comprimir um típico Cabeçalho TCP / IP de 40 bytes para uma média de 3 bytes. O truque para esta method é sugerido na Figura 6.52. Muitos dos campos do cabeçalho não mudam do pacote para empacotar. Não há necessidade, por exemplo, de enviar o mesmo IP TTL ou o mesmo Números de porta TCP em cada pacote. Eles podem ser omitidos no envio lado do link e preenchido no lado receptor.

Da mesma forma, outros campos mudam de maneira previsível. Por exemplo, barrar perda, o número de sequência TCP avança com os dados. Nestes casos, o receiver pode prever o valor provável. O número real só precisa ser carregado quando é diferente do esperado. Mesmo assim, pode ser transportado como um pequeno mudança do valor anterior, como quando o número de confirmação aumenta quando novos dados são recebidos na direção reversa.

Com a compressão de cabeçalho, é possível ter cabeçalhos simples em camadas superiores protocolos e codificações compactas em links de baixa largura de banda. **ROHC** (**RObust Compressão de cabeçalho**) é uma versão moderna da compressão de cabeçalho que é definida

Página 619

SEC. 6,6

PROBLEMAS DE DESEMPENHO

595

como uma estrutura no RFC 5795. Ele é projetado para tolerar a perda que pode ocorrer em links sem fio. Existe um perfil para cada conjunto de protocolos a serem compactados, como como IP / UDP / RTP. Cabeçalhos compactados são transportados por referência a um contexto, que é essencialmente uma conexão; campos de cabeçalho podem ser facilmente previstos para pacotes de a mesma conexão, mas não para pacotes de conexões diferentes. Em operação típica ação, o ROHC reduz os cabeçalhos IP / UDP / RTP de 40 bytes para 1 a 3 bytes.

Embora a compressão do cabeçalho seja principalmente voltada para a redução das necessidades de largura de banda,

também pode ser útil para reduzir o atraso. O atraso é composto de atraso de propagação, que é fixo, dado um caminho de rede e atraso de transmissão, que depende de a largura de banda e a quantidade de dados a serem enviados. Por exemplo, um link de 1 Mbps envia 1

bit em 1 μ seg. No caso de mídia em redes sem fio, a rede é relativa lentos, então o atraso de transmissão pode ser um fator importante no atraso geral e na con atrasos consistentemente baixos são importantes para a qualidade do serviço.

A compressão do cabeçalho pode ajudar, reduzindo a quantidade de dados que é enviada, e portanto, reduzindo o atraso de transmissão. O mesmo efeito pode ser alcançado enviando pacotes menores. Isso vai trocar o aumento da sobrecarga de software por uma redução de atraso da missão. Observe que outra fonte potencial de atraso é o atraso da fila para ac cessar o link sem fio. Isso também pode ser significativo porque os links sem fio são frequentemente muito usado como o recurso limitado em uma rede. Neste caso, o link sem fio deve ter mecanismos de qualidade de serviço que forneçam baixo atraso aos pacotes em tempo real. A compactação do cabeçalho por si só não é suficiente.

6.6.6 Protocolos para Redes Long Fat

Desde a década de 1990, existem redes gigabit que transmitem dados por grandes distâncias. Por causa da combinação de uma rede rápida, ou "tubo gordo", e longo atraso, essas redes são chamadas de **redes longas e gordas**. Quando essas redes surgiram, a primeira reação das pessoas foi usar os protocolos existentes nelas, mas vários nossos problemas surgiram rapidamente. Nesta seção, discutiremos alguns dos problemas com o aumento da velocidade e do atraso dos protocolos de rede.

O primeiro problema é que muitos protocolos usam números de sequência de 32 bits. Quando a Internet começou, as linhas entre os roteadores eram principalmente linhas alugadas de 56 kbps, então

um host explodindo a toda velocidade levou mais de 1 semana para percorrer a sequência números. Para os projetistas do TCP, 2³² foi uma aproximação bastante decente do infinito porque havia pouco perigo de pacotes antigos ainda estarem por perto uma semana depois de foram transmitidos. Com Ethernet de 10 Mbps, o tempo de empacotamento tornou-se 57 minutos,

muito mais curto, mas ainda administrável. Com uma Ethernet de 1 Gbps enviando dados para a Internet, o tempo de embrulho é de cerca de 34 segundos, bem abaixo do máximo de 120 segundos tempo de vida do pacote na Internet. De repente, 2³² não é tão bom um aproximação ao infinito, uma vez que um remetente rápido pode percorrer o espaço de sequência enquanto os pacotes antigos ainda existem. O problema é que muitos designers de protocolo simplesmente assumiram, sem declarar isso, que o tempo necessário para usar todo o espaço da sequência excederia em muito

Página 620

596

A CAMADA DE TRANSPORTE INDIVÍDUO. 6

a vida útil máxima do pacote. Consequentemente, não havia necessidade nem mesmo de se preocupar sobre o problema de duplicatas antigas ainda existentes quando os números de sequência enrolado. Em velocidades de gigabit, essa suposição não declarada falha. Felizmente, provou ser possível estender o número de sequência eficaz, tratando o tempo carimbo que pode ser transportado como uma opção no cabeçalho TCP de cada pacote como o bits de alta ordem. Este mecanismo é denominado PAWS (Protection Against Wrapped Números de sequência) e é descrito no RFC 1323.

Um segundo problema é que o tamanho da janela de controle de fluxo deve ser muito aumentou. Considere, por exemplo, o envio de um burst de 64 KB de dados de San Diego para Boston para preencher o buffer de 64 KB do receptor. Suponha que o link seja 1 Gbps e o atraso da velocidade da luz na fibra unidirecional é de 20 mseg. Inicialmente, em $t = 0$, o tubo está vazio, conforme ilustrado na Fig. 6-54 (a). Apenas 500 μ s depois, na Fig. 6-54 (b), todos os segmentos estão fora da fibra. O segmento principal agora será onde nas proximidades de Brawley, ainda no sul da Califórnia. No entanto, o transmissor deve parar até obter uma atualização da janela.

(uma)

(b)

(c)

(d)

Dados

Reconhecimentos

Figura 6-54. O estado de transmissão de 1 Mbit de San Diego para Boston. (a) Em $t = 0$. (b) Após 500 μ seg. (c) Após 20 mseg. (d) Após 40 mseg.

Após 20 ms, o segmento principal atinge Boston, conforme mostrado na Figura 6-54 (c), e é reconhecido. Finalmente, 40 ms após o início, o primeiro reconhecimento obtém

Página 621

SEC. 6,6

PROBLEMAS DE DESEMPENHO

597

de volta ao remetente e a segunda rajada pode ser transmitida. Desde a transmissão linha foi usada para 1,25 ms de 100, a eficiência é de cerca de 1,25%. Esta situação é típica de protocolos mais antigos rodando em linhas de gigabit.

Uma quantidade útil a se ter em mente ao analisar o desempenho da rede é o **produto de atraso de largura de banda**. É obtido multiplicando a largura de banda (em bits / s) pelo tempo de retardo de ida e volta (em segundos). O produto é a capacidade do canalizar do emissor para o receptor e vice-versa (em bits).

Para o exemplo da Figura 6.54, o produto do atraso da largura de banda é de 40 milhões de bits.

Em outras palavras, o remetente teria que transmitir uma rajada de 40 milhões de bits para ser capaz de continuar a toda velocidade até a primeira confirmação voltar. Leva essa quantidade de bits para encher o tubo (em ambas as direções). É por isso que uma explosão de meio

milhão de bits atinge apenas 1,25% de eficiência: é apenas 1,25% da capacidade do tubo ity.

A conclusão que pode ser tirada aqui é que para um bom desempenho, o re-a janela do ceiver deve ser pelo menos tão grande quanto o produto de atraso de largura de banda, e

de preferência um pouco maior, pois o receptor pode não responder instantaneamente. Para linha de gigabit transcontinental, pelo menos 5 MB são necessários.

Um terceiro problema relacionado é que esquemas de retransmissão simples, como o protocolo go-back-n, desempenho insatisfatório em linhas com um grande produto de atraso de largura de banda.

Considere o link transcontinental de 1 Gbps com um tempo de transmissão de ida e volta de 40 mseg. Um remetente pode transmitir 5 MB em uma viagem de ida e volta. Se um erro for detectado,

será de 40 ms antes de o remetente ser informado sobre isso. Se go-back-n for usado, o remetente terá que retransmitir não apenas o pacote ruim, mas também o valor de 5 MB de pacotes que veio depois. Claramente, isso é um grande desperdício de recursos. Mais complexos protocolos como a repetição seletiva são necessários.

Um quarto problema é que as linhas de gigabit são fundamentalmente diferentes de megabit as linhas de bits nessas linhas de gigabit longas são limitadas por atraso, em vez de pela largura de banda.

Na Fig. 6-55, mostramos o tempo que leva para transferir um arquivo de 1 Mbit 4000 km em várias velocidades de transmissão. Em velocidades de até 1 Mbps, o tempo de transmissão é dominado pela taxa na qual os bits podem ser enviados. Por 1 Gbps, o atraso de ida e volta de 40 ms domina o 1 ms que leva para colocar os bits na fibra. Aumentos adicionais em a largura de banda quase não tem efeito algum.

A Figura 6-55 tem implicações infelizes para os protocolos de rede. Isso diz que protocolos de parar e esperar, como RPC, têm um limite superior inerente em seus desempenhos. Este limite é ditado pela velocidade da luz. Nenhuma quantidade de tecnologia ou progresso da óptica sempre melhorará as coisas (novas leis da física ajudariam, Apesar). A menos que algum outro uso possa ser encontrado para uma linha gigabit enquanto um host é

esperando por uma resposta, a linha gigabit não é melhor que uma linha megabit, apenas mais expensivo.

Um quinto problema é que as velocidades de comunicação melhoraram mais rápido do que as velocidades de colocação. (Nota para engenheiros de computação: saia e vença os comunicadores engenheiros de instalação! Contamos com você.) Na década de 1970, a ARPANET funcionava a 56 kbps e tinha computadores que funcionavam a cerca de 1 MIPS. Compare esses números com

Página 622

598

A CAMADA DE TRANSPORTE INDIVÍDUO. 6

1000 s
100 s
10 s
1 segundo
100 mseg
10 mseg
1 mseg
Arquivo
transferir
Tempo
Taxa de dados (bps)
10³
10⁴
10⁵
10⁶
10⁷
10⁸
10⁹
10¹⁰
10¹¹
10¹²

Figura 6-55. É hora de transferir e reconhecer um arquivo de 1 Mbit ao longo de 4000 km linha.

Computadores de 1000 MIPS trocando pacotes em uma linha de 1 Gbps. O número de estruturas por byte diminuiu em mais de um fator de 10. Os números exatos são discutíveis dependendo das datas e cenários, mas a conclusão é esta: há

menos tempo disponível para processamento de protocolo do que costumava ser, então os protocolos devem se tornar mais simples.

Passemos agora dos problemas para maneiras de lidar com eles. O básico O princípio que todos os designers de rede de alta velocidade devem aprender de cor é:

Projete para velocidade, não para otimização de largura de banda.

Os protocolos antigos costumavam ser projetados para minimizar o número de bits na transmissão, freqüentemente usando pequenos campos e agrupando-os em bytes e palavras.

Essa preocupação ainda é válida para redes sem fio, mas não para redes gigabit.

O processamento de protocolo é o problema, portanto, os protocolos devem ser projetados para minimizar

isto. Os projetistas do IPv6 entenderam claramente esse princípio.

Uma maneira tentadora de acelerar é construir interfaces de rede rápidas em hardware. o dificuldade com esta estratégia é que, a menos que o protocolo seja extremamente simples, difícil ware significa apenas uma placa plug-in com uma segunda CPU e seu próprio programa. Para certifique-se de que o coprocessador da rede é mais barato do que a CPU principal, geralmente é um chip mais lento. A consequência deste design é que na maior parte do tempo o principal A CPU (rápida) está ociosa esperando que a segunda CPU (lenta) faça o trabalho crítico. Isto é um mito pensar que a CPU principal tem outro trabalho a fazer enquanto espera. Pele- além disso, quando duas CPUs de uso geral se comunicam, as condições de corrida podem ocorrer agora, então protocolos elaborados são necessários entre os dois processadores para sincronizar

Página 623

SEC. 6,6

PROBLEMAS DE DESEMPENHO

599

-los corretamente e evitar corridas. Normalmente, a melhor abordagem é fazer o proto- cols simples e ter a CPU principal fazendo o trabalho.

O layout do pacote é uma consideração importante em redes gigabit. O cabecalho deve conter o mínimo de campos possível, para reduzir o tempo de processamento, e estes os campos devem ser grandes o suficiente para fazer o trabalho e ser alinhados por palavra para processamento rápido.

Neste contexto, " grande o suficiente " significa que problemas como números de sequência enrolando enquanto os pacotes antigos ainda existem, os receptores sendo incapazes de anunciar espaço de janela suficiente porque o campo da janela é muito pequeno, etc. não ocorrem.

O tamanho máximo dos dados deve ser grande, para reduzir a sobrecarga de software e mit operação eficiente. 1500 bytes é muito pequeno para redes de alta velocidade, o que é por que a Ethernet gigabit oferece suporte a quadros jumbo de até 9 KB e suporte a IPv6 pacotes de jumbogram com mais de 64 KB.

Vejamos agora a questão do feedback em protocolos de alta velocidade. Devido ao loop de atraso (relativamente) longo, o feedback deve ser evitado: leva muito tempo para o receptor para sinalizar o remetente. Um exemplo de feedback é governar a transmissão taxa de concentração usando um protocolo de janela deslizante. Protocolos futuros podem mudar para

protocolos baseados em taxas para evitar os (longos) atrasos inerentes ao envio do receptor atualizações de janela para o remetente. Nesse protocolo, o remetente pode enviar tudo o que quiser para, desde que não envie mais rápido do que alguma taxa que o remetente e o destinatário têm previamente acordado.

Um segundo exemplo de feedback é o algoritmo de início lento de Jacobson. Este algo-

O rithm faz várias investigações para ver o quanto a rede pode suportar. Com redes de alta velocidade, fazendo meia dúzia de pequenas sondagens para ver como a rede trabalha responde desperdiça uma grande quantidade de largura de banda. Um esquema mais eficiente é

faça com que o emissor, o receptor e a rede reservem os recursos necessários em con- tempo de configuração da conexão. Reservar recursos com antecedência também tem a vantagem de ma-

torna mais fácil reduzir o jitter. Em suma, ir para altas velocidades empurra inexoravelmente o

design para operação orientada a conexão, ou algo bastante próximo a isso. Outro recurso valioso é a capacidade de enviar uma quantidade normal de dados junto com a solicitação de conexão. Desta forma, um tempo de ida e volta pode ser salvo.

6.7 REDE TOLERANTE DE ATRASO

Terminaremos este capítulo descrevendo um novo tipo de transporte que pode dia ser um componente importante da Internet. TCP e muitos outros meios de transporte tocols são baseados na suposição de que o remetente e o receptor são contínuos firmemente conectado por algum caminho de trabalho, ou então o protocolo falha e os dados não podem ser entregue. Em algumas redes, geralmente não há um caminho de ponta a ponta. Um exemplo é um rede espacial à medida que os satélites LEO (Low-Earth Orbit) entram e saem do alcance de estações terrestres. Um determinado satélite pode ser capaz de se comunicar com uma estação terrestre apenas em momentos específicos, e dois satélites podem nunca ser capazes de se comunicar com uns aos outros a qualquer momento, mesmo através de uma estação terrestre, porque um dos satélites

Página 624

600

A CAMADA DE TRANSPORTE INDIVÍDUO. 6

pode sempre estar fora do alcance. Outros exemplos de redes envolvem submarinos, ônibus, telefones celulares e outros dispositivos com computadores para os quais há conectividade devido à mobilidade ou condições extremas.

Nessas redes ocasionalmente conectadas, os dados ainda podem ser comunicados por armazená-los nos nós e encaminhá-los posteriormente quando houver um link funcionando. Essa técnica é chamada de **troca de mensagens**. Eventualmente, os dados serão retransmitidos para o destino. Uma rede cuja arquitetura é baseada nesta abordagem é chamada ed uma **DTN (Rede Tolerante a Delay ou Rede Tolerante a Disrupções)**.

O trabalho com DTNs começou em 2002, quando a IETF criou um grupo de pesquisa sobre o tema. A inspiração para DTNs veio de uma fonte improvável: esforços para enviar pacotes no espaço. As redes espaciais devem lidar com comunicação intermitente e atrasos muito longos. Kevin Fall observou que as idéias para esses interplanetários ternets podem ser aplicadas a redes na Terra nas quais a conectividade intermitente era a norma (outono, 2003). Este modelo dá uma generalização útil do Inter-rede na qual armazenamento e atrasos podem ocorrer durante a comunicação. Entrega de dados é semelhante à entrega no sistema postal, ou correio eletrônico, em vez de pacote comutação em roteadores.

Desde 2002, a arquitetura DTN foi refinada, e as aplicações do O modelo DTN cresceu. Como um aplicativo mainstream, considere grandes conjuntos de dados de muitos terabytes que são produzidos por experimentos científicos, eventos de mídia ou Serviços baseados na web e precisam ser copiados para datacenters em locais diferentes em todo o mundo. Os operadores gostariam de enviar este tráfego em massa fora dos horários de pico

para fazer uso da largura de banda que já foi paga, mas não está sendo usada, e estão dispostos a tolerar algum atraso. É como fazer backups à noite, quando outros os aplicativos não estão fazendo uso intenso da rede. O problema é que, para serviços globais, os horários fora de pico são diferentes em locais ao redor do mundo.

Pode haver pouca sobreposição nos momentos em que os datacenters em Boston e Perth têm largura de banda de rede fora do pico porque a noite para uma cidade é dia para a outra. No entanto, os modelos DTN permitem armazenamento e atrasos durante a transferência. Com neste modelo, torna-se possível enviar o conjunto de dados de Boston para Amsterdã usando largura de banda fora do pico, pois as cidades têm fusos horários de apenas 6 horas separados. O conjunto de dados é então armazenado em Amsterdã até que haja largura de banda fora do pico

entre Amsterdã e Perth. Em seguida, ele é enviado a Perth para concluir a transferência.

Laoutaris et al. (2009) estudaram este modelo e descobriram que ele pode fornecer sub-capacidade substancial a baixo custo, e que o uso de um modelo DTN muitas vezes dobra isso

capacidade em comparação com um modelo ponta a ponta tradicional.

A seguir, descreveremos a arquitetura e os protocolos IETF DTN.

6.7.1 Arquitetura DTN

A principal suposição na Internet de que as DTNs procuram relaxar é que um fim o caminho final entre uma origem e um destino existe durante toda a duração de um sessão de comunicação. Quando este não é o caso, os protocolos normais da Internet

Página 625

SEC. 6,7

REDE TOLERANTE DE ATRASO

601

falhou. DTNs contornam a falta de conectividade ponta a ponta com uma arquitetura que se baseia na troca de mensagens, conforme mostrado na Figura 6-56. Também se destina a tolera links com baixa confiabilidade e grandes atrasos. A arquitetura é especificada em RFC 4838.

Contato
(link de trabalho)
Armazenado
agrupar
Fonte
Armazenamento
Enviei
agrupar
DTN
nó
Link intermitente
(não está funcionando)
Destino

Figura 6-56. Arquitetura de rede tolerante a atrasos.

Na terminologia DTN, uma mensagem é chamada de **pacote**. Nós DTN são equipados com armazenamento, geralmente armazenamento persistente, como um disco ou memória flash. Eles armazene os pacotes até que os links estejam disponíveis e, em seguida, encaminhe os pacotes. Os links

trabalhar de forma intermitente. A Fig. 6-56 mostra cinco links intermitentes que atualmente não estão funcionando, e dois links que estão funcionando. Um link ativo é chamado de **contato**.

A Fig. 6-56 também mostra pacotes armazenados em dois nós DTN aguardando contatos para enviar os pacotes em diante. Desta forma, os pacotes são retransmitidos por meio de contatos do origem ao destino.

O armazenamento e encaminhamento de pacotes em nós DTN soa semelhante ao enfileiramento e encaminhamento de pacotes em roteadores, mas existem diferenças qualitativas ferências. Em roteadores na Internet, o enfileiramento ocorre por milissegundos ou no máximo segundos. Em nós DTN, os pacotes podem ser armazenados por horas, até que um ônibus chegue em cidade, enquanto um avião completa um vôo, até que um nó sensor colete o suficiente energia solar para funcionar, até que um computador em modo de espera acorde, e assim por diante. Estes ex-

amplos também apontam para uma segunda diferença, que é que os nós podem se mover (com um ônibus ou avião) enquanto eles mantêm os dados armazenados, e este movimento pode até ser uma chave

parte da entrega de dados. Os roteadores na Internet não podem se mover. O todo processo de movimentação de pacotes pode ser mais conhecido como "transporte de armazenamento".

Como exemplo, considere o cenário mostrado na Figura 6-57, que foi o primeiro uso de protocolos DTN no espaço (Wood et al., 2008). A fonte dos pacotes é um LEO satélite que está gravando imagens da Terra como parte do Constel de Monitoramento de Desastres instalação de satélites. As imagens devem ser devolvidas ao ponto de coleta. Contudo, o satélite tem apenas contato intermitente com três estações terrestres enquanto orbita o Terra. Ele entra em contato com cada estação terrestre sucessivamente. Cada um dos satélites, estações terrestres e pontos de coleta atuam como um nó DTN. A cada contato, um

Página 626

602

A CAMADA DE TRANSPORTE INDIVÍDUO. 6

pacote (ou uma parte de um pacote) é enviado para uma estação terrestre. Os pacotes são então enviado através de uma rede terrestre de backhaul para o ponto de coleta para completar o transferir.

Link intermitente
(não está funcionando)
Armazenamento em
Nós DTN
Satélite
Contato
(link de trabalho)
Agrupar
Terra
estação
Ponto de Coleta

Figura 6-57. Uso de um DTN no espaço.

A principal vantagem da arquitetura DTN neste exemplo é que ela naturalmente encaixa perfeitamente na situação do satélite precisando armazenar imagens porque não há conectividade no momento em que a imagem é tirada. Existem mais duas vantagens.

Primeiro, pode não haver um único contato por tempo suficiente para enviar as imagens. Contudo, eles podem ser espalhados pelos contatos com três estações terrestres. Em segundo lugar, o uso do link entre o satélite e a estação terrestre é desacoplado do link sobre a rede de backhaul. Isso significa que o download do satélite não é limitado por uma ligação terrestre lenta. Ele pode prosseguir a toda velocidade, com o pacote armazenado na estação terrestre até que possa ser retransmitida ao ponto de coleta.

Uma questão importante que não é especificada pela arquitetura é como encontrar boas rotas através de nós DTN. Uma rota neste caminho para usar. Boas rotas dependem da natureza da arquitetura descreve quando enviar dados e também quais contatos.

Alguns contatos são conhecidos com antecedência. Um bom exemplo é o movimento de corpos celestes no exemplo do espaço. Para o experimento espacial, era conhecido antes do tempo em que os contatos ocorreriam, os intervalos de contato variaram de 5 a 14 minutos por passagem com cada estação terrestre, e que a capacidade do downlink era de 8,134 Mbps. Dado esse conhecimento, o transporte de um feixe de imagens pode ser planejado com antecedência.

Em outros casos, os contatos podem ser previstos, mas com menos certeza. Exemplos incluem ônibus que fazem contato uns com os outros de maneiras principalmente regulares, devido a um horário, mas com alguma variação, e os horários e quantidade de banda fora de pico largura em redes ISP, que são previstas a partir de dados anteriores. No outro extremo, os contatos são ocasionais e aleatórios. Um exemplo é o transporte de dados do usuário

Página 627

SEC. 6,7 REDE TOLERANTE DE ATRASO

603

para o usuário em telefones celulares, dependendo de quais usuários fazem contato uns com os outros

durante o dia. Quando há imprevisibilidade nos contatos, uma estratégia de roteamento é para enviar cópias do pacote por caminhos diferentes na esperança de que um dos pacotes seja entregue ao destino antes que o tempo de vida seja alcançado.

6.7.2 O Protocolo de Bundle

Para dar uma olhada mais de perto na operação de DTNs, veremos agora o IETF protocolos. DTNs são um tipo emergente de rede, e DTNs experimentais têm usaram protocolos diferentes, pois não há exigência de que os protocolos IETF sejam usados. No entanto, eles são pelo menos um bom lugar para começar e destacar muitos dos assuntos chave.

A pilha do protocolo DTN é mostrada na Figura 6.58. O protocolo principal é o **Bundle protocolo dle**, que é especificado na RFC 5050. É responsável por aceitar mensagens do aplicativo e enviá-las como um ou mais pacotes via loja-

operações de transporte para o nó DTN de destino. Também é aparente de Fig. 6-58 mostra que o protocolo Bundle é executado acima do nível de TCP / IP. Em outras palavras, O TCP / IP pode ser usado em cada contato para mover pacotes entre nós DTN. Este posicionamento levanta a questão de saber se o protocolo Bundle é um transporte protocolo de camada ou um protocolo de camada de aplicativo. Assim como com a RTP, tomamos a posição que, apesar de rodar sobre um protocolo de transporte, o protocolo Bundle é fornecendo um serviço de transporte para muitos aplicativos diferentes, e assim cobrimos DTNs neste capítulo.

```
Inscrição
Protocolo de Pacote
Camada de convergência
TCP / IP
Internet
...
De outros
Internet
Camada de convergência
Superior
camadas
DTN
camada
Mais baixo
camadas
```

Figura 6-58. Pilha de protocolo de rede tolerante a atrasos.

Na Figura 6-58, vemos que o protocolo Bundle pode ser executado sobre outros tipos de protocolos como UDP, ou mesmo outros tipos de internet. Por exemplo, em um espaço rede os links podem ter atrasos muito longos. O tempo de ida e volta entre a Terra e Marte pode facilmente ter 20 minutos, dependendo da posição relativa dos planetas. Imagine o quanto bem os reconhecimentos e retransmissões TCP funcionariam sobre esse link, especialmente para mensagens relativamente curtas. Não está nada bem. Em vez de,

Página 628

604

A CAMADA DE TRANSPORTE

INDIVÍDUO. 6

outro protocolo que usa códigos de correção de erros pode ser usado. Ou na rede do sensor trabalhos que têm muitos recursos limitados, um protocolo mais leve que o TCP pode ser usado.

Uma vez que o protocolo Bundle é fixo, ele se destina a funcionar em uma variedade de transportes, deve haver uma lacuna de funcionalidade entre os protocolos. Essa lacuna é a razão para a inclusão de uma camada de convergência na Figura 6.58. O convergente é apenas uma camada de cola que corresponde às interfaces dos protocolos que junta-se. Por definição, há uma camada de convergência diferente para cada diferente transporte de camada. Camadas de convergência são comumente encontradas em padrões para unir novos e protocolos existentes.

O formato das mensagens do protocolo Bundle é mostrado na Figura 6.59. Os diferentes campos nessas mensagens nos contam alguns dos principais problemas que são tratados pelo Protocolo de pacote.

Bits	
7	
7	
Tipo	
Bloco primário	
Bloco de carga útil	
Blocos opcionais	
Comprimento	
Dados	
Ver. Bandeiras	
Dest.	
Bandeiras	
Source Report Custodian Creation Lifetime Dictionary	
8	
variável	
Status	
relatório	
Classe de	
serviço	
Geral	
Bits	
8	
6	
variável	
6	
6	
20	

Figura 6-59. Formato de mensagem de protocolo do pacote.

Cada mensagem consiste em um bloco primário, que pode ser pensado como um cabeçalho ou, um bloco de carga útil para os dados e, opcionalmente, outros blocos, por exemplo, para transportar parâmetros de segurança. O bloco primário começa com um campo de *versão* (atualmente 6) seguido por um campo *Flags*. Entre outras funções, os sinalizadores codificam uma classe de serviço para permitir que uma fonte marque seus pacotes como de maior ou menor prioridade, e outros handling solicitações, como se o destino deve reconhecer o pacote.

Em seguida, vêm os endereços, que destacam três partes interessantes do design. Como bem como um campo de identificador de *destino* e *fonte*, há um identificador de *custodiante*. O guardião é o responsável por garantir a entrega do maço. No Internet, o nó de origem é geralmente o guardião, pois é o nó que retransmite se os dados não forem finalmente entregues ao destino. No entanto, em um DTN, o nó de origem pode nem sempre estar conectado e pode não ter como saber se os dados foram entregues. DTNs lidam com este problema usando a noção de **transferência de custódia**, na qual outro nó, mais próximo do destino, pode assumir a responsabilidade de ver os dados entregues com segurança. Por exemplo, se um bolo é armazenado em um avião para encaminhamento em um momento e local posterior, o avião pode se tornar o guardião do pacote.

Página 629

SEC. 6,7

REDE TOLERANTE DE ATRASO

605

O segundo aspecto interessante é que esses identificadores *não* são endereços IP. Estar porque o protocolo Bundle se destina a funcionar em uma variedade de transportes e internets, ele define seus próprios identificadores. Esses identificadores são realmente mais parecidos com nomes de alto nível, como URLs de páginas da web, do que endereços de baixo nível, como IP endereços. Eles fornecem aos DTNs um aspecto do roteamento no nível do aplicativo, como e-mail entrega ou distribuição de atualizações de software.

O terceiro aspecto interessante é a maneira como os identificadores são codificados. Há sim também um identificador de *relatório* para mensagens de diagnóstico. Todos os identificadores são codificados

como referências a um campo de *dicionário de comprimento variável*. Isso fornece compressão quando o custodiante ou os nós do relatório são iguais à origem ou ao destino.

Na verdade, muito do formato da mensagem foi projetado com extensibilidade e eficiência em mente usando uma representação compacta de campos de comprimento variável.

A representação compacta é importante para links sem fio e recursos nós restritos, como em uma rede de sensores.

Em seguida, vem um campo de *Criação* carregando o momento em que o pacote foi criado - ed, junto com um número de sequência da fonte para pedido, mais um *Lifetime* campo que informa a hora em que os dados do pacote não são mais úteis. Esses campos existem porque os dados podem ser armazenados por um longo período em nós DTN e deve haver ser uma forma de remover dados obsoletos da rede. Ao contrário da Internet, eles voltam exigem que os nós DTN tenham relógios fracamente sincronizados.

O bloco principal é completado com o campo *Dicionário*. Então vem o bloco de carga útil. Este bloco começa com um pequeno campo *Tipo* que o identifica como um pagamento

carregar, seguido por um pequeno conjunto de *sinalizadores* que descrevem as opções de processamento. Então

vem o campo *Dados*, precedido por um campo *Comprimento*. Finalmente, pode haver outro, op-blocos tradicionais, como um bloco que carrega parâmetros de segurança.

Muitos aspectos das DTNs estão sendo explorados na comunidade de pesquisa. Boa as estratégias de encaminhamento dependem da natureza dos contatos, como foi mencionado acima. O armazenamento de dados na rede levanta outros problemas. Agora controle de congestionamento

deve considerar o armazenamento em nós como outro tipo de recurso que pode ser esgotado. A falta de comunicação ponta a ponta também agrava os problemas de segurança. Antes um nó DTN assume a custódia de um pacote, ele pode querer saber se o remetente é autorizado a usar a rede e que o pacote é provavelmente desejado pelo destino nação. As soluções para esses problemas dependerão do tipo de DTN, como rede espacial obras são diferentes de redes de sensores.

6.8 RESUMO

A camada de transporte é a chave para entender os protocolos em camadas. Fornece vários serviços, o mais importante dos quais é um sistema de ponta a ponta, confiável, fluxo de bytes orientado para a conexão do emissor para o receptor. É acessado por meio de serviço-primitivas que permitem o estabelecimento, uso e liberação de conexões. UMA a interface da camada de transporte comum é aquela fornecida pelos soquetes Berkeley.

Página 630

606

A CAMADA DE TRANSPORTE INDIVÍDUO. 6

Os protocolos de transporte devem ser capazes de fazer o gerenciamento de conexão em vez de redes responsáveis. O estabelecimento da conexão é complicado pela existência de de-pacotes duplicados colocados que podem reaparecer em momentos inoportunos. Lidar com para eles, são necessários handshakes de três vias para estabelecer conexões. Liberando um conexão é mais fácil do que estabelecer uma, mas ainda está longe de ser trivial devido ao problema de dois exércitos.

Mesmo quando a camada de rede é totalmente confiável, a camada de transporte tem muito trabalho a fazer. Deve lidar com todos os primitivos de serviço, gerenciar conexões e temporizadores, alocar largura de banda com controle de congestionamento e executar uma variável

janela deslizante dimensionada para controle de fluxo.

O controle de congestionamento deve alocar toda a largura de banda disponível entre a concorrência fluindo de forma justa e deve rastrear as mudanças no uso da rede.

A lei de controle AIMD converge para uma alocação justa e eficiente.

A Internet tem dois protocolos de transporte principais: UDP e TCP. UDP é um protocolo sem conexão que é principalmente um wrapper para pacotes IP com o adicional recurso de multiplexação e demultiplexação de vários processos usando um único IP endereço. O UDP pode ser usado para interações cliente-servidor, por exemplo, usando RPC. Ele também pode ser usado para construir protocolos em tempo real, como RTP.

O principal protocolo de transporte da Internet é o TCP. Ele fornece um confiável e bidirecional fluxo de bytes controlado por congestionamento com um cabeçalho de 20 bytes em todos os segmentos.

Muito trabalho foi feito para otimizar o desempenho do TCP, usando algoritmos de Nagle, Clark, Jacobson, Karn e outros.

O desempenho da rede é normalmente dominado pelo protocolo e processo de segmento sobrecarregar, e esta situação piora em velocidades mais altas. Protocolos devem ser projetados para minimizar o número de segmentos e trabalhar para uma grande largura de banda caminhos de atraso. Para redes gigabit, protocolos simples e processamento simplificado são chamados.

A rede tolerante a atrasos fornece um serviço de entrega em redes que têm conectividade ocasional ou longos atrasos nos links. Nós intermediários armazenar, transportar e encaminhar pacotes de informações para que sejam eventualmente entregues,

mesmo se não houver um caminho de trabalho do emissor ao receptor a qualquer momento.

PROBLEMAS

1. Em nosso exemplo de primitivas de transporte da Figura 6.2, LISTEN é uma chamada de bloqueio. É isto estritamente necessário? Caso contrário, explique como uma primitiva não bloqueante pode ser usada. o que vantagem que isso teria sobre o esquema descrito no texto?

2. As primitivas do serviço de transporte assumem assimetria entre os dois pontos finais durante estabelecimento de conexão, uma extremidade (servidor) executa LISTEN enquanto a outra extremidade (cliente) executa CONNECT. No entanto, em aplicativos ponto a ponto, como o compartilhamento de arquivos

INDIVÍDUO. 6

PROBLEMAS

607

sistemas, por exemplo, BitTorrent, todos os pontos finais são pares. Não há servidor ou função de cliente nacionalidade. Como as primitivas de serviço de transporte podem ser usadas para construir tal ponto a ponto formulários?

3. No modelo subjacente da Figura 6-4, assume-se que os pacotes podem ser perdidos pela rede. camada de trabalho e, portanto, deve ser reconhecido individualmente. Suponha que a rede camada é 100 por cento confiável e nunca perde pacotes. Quais mudanças, se houver, são necessário para Fig. 6-4?
4. Em ambas as partes da Fig. 6-6, há um comentário de que o valor de *SERVER PORT* deve ser o mesmo no cliente e no servidor. Por que isso é tão importante?
5. No exemplo do Servidor de Arquivos da Internet (Figura 6-6), o sistema *connect ()* pode chamar no falha do cliente por qualquer motivo diferente de fila de escuta cheia no servidor? Assuma isso a rede é perfeita.
6. Um critério para decidir se deve ter um servidor ativo o tempo todo ou iniciá-lo demanda usando um servidor de processo é a frequência com que o serviço fornecido é usado. pode você pensa em algum outro critério para tomar essa decisão?
7. Suponha que o esquema baseado em relógio para gerar números de sequência iniciais seja usado com um contador de relógio de 15 bits. O relógio bate uma vez a cada 100 ms, e o máximo a vida útil máxima do pacote é de 60 segundos. Com que frequência a resincronização precisa ocorrer (a) no pior caso?
(b) quando os dados consomem 240 números de sequência / min?
8. Por que o tempo de vida máximo do pacote, T , tem que ser grande o suficiente para garantir que não apenas o pacote, mas também seus reconhecimentos desapareceram?
9. Imagine que um handshake de duas vias, em vez de um handshake de três vias, fosse usado para definir conexões. Em outras palavras, a terceira mensagem não era necessária. São impasses agora é possível? Dê um exemplo ou mostre que não existe.
10. Imagine um problema n -army generalizado , em que a concordância de quaisquer dois dos exércitos é suficiente para a vitória. Existe um protocolo que permite que o azul vença?
11. Considere o problema de recuperação de travamentos do host (ou seja, Figura 6-18). Se o intervalo entre escrever e enviar uma confirmação, ou vice-versa, pode ser feita relativamente pequeno, quais são as duas melhores estratégias emissor-receptor para minimizar o chance de uma falha de protocolo?
12. Na Figura 6-20, suponha que um novo fluxo E seja adicionado, levando um caminho de $R1$ a $R2$ e $R6$. Como a alocação de largura de banda máxima-mínima muda para os cinco fluxos?
13. Discuta as vantagens e desvantagens dos protocolos de crédito versus janela deslizante.
14. Algumas outras políticas para justiça no controle de congestionamento são Additive Aumento Aditivo Diminuir (AIAD), Aumentar Multiplicativo e Diminuir Aditivo (MIAD) e Multiplicar Aumento cativo Redução multiplicativa (MIMD). Discuta essas três políticas em termos de convergência e estabilidade.
15. Por que existe o UDP? Não teria sido suficiente apenas permitir que os processos do usuário enviassem pacotes IP brutos?

608

A CAMADA DE TRANSPORTE

INDIVÍDUO. 6

16. Considere um protocolo de nível de aplicativo simples construído em cima do UDP que permite a um cliente recuperar um arquivo de um servidor remoto residente em um endereço conhecido. O cliente primeiro envia uma solicitação com um nome de arquivo e o servidor responde com uma sequência de dados pacotes contendo diferentes partes do arquivo solicitado. Para garantir confiabilidade e entrega sequenciada, cliente e servidor usam um protocolo stop-and-wait. Ignorando o óbvio performance, você vê um problema com este protocolo? Pense com cuidado sobre a possibilidade de travamento de processos.
17. Um cliente envia uma solicitação de 128 bytes para um servidor localizado a 100 km de distância em um 1 gigabit fibra ótica. Qual é a eficiência da linha durante a chamada de procedimento remoto?
18. Considere a situação do problema anterior novamente. Calcule o mínimo possível tempo de resposta para a linha de 1 Gbps fornecida e para uma linha de 1 Mbps. Que conclusão você pode desenhar?
19. Tanto UDP quanto TCP usam números de porta para identificar a entidade de destino ao entregar

ing uma mensagem. Dê duas razões pelas quais esses protocolos inventaram um novo ID abstrato (porta números), em vez de usar IDs de processo, que já existiam quando esses protocolos foram projetados.

20. Várias implementações de RPC fornecem uma opção para o cliente usar o RPC implementado sobre UDP ou RPC implementado sobre TCP. Em que condições o cliente preferirá usar RPC sobre UDP e em que condições ele preferirá usar RPC sobre TCP?

21. Considere duas redes, N_1 e N_2 , que têm o mesmo atraso médio entre um fonte A e um destino D . Em N_1 , o atraso experimentado por diferentes pacotes é uniformemente distribuído com atraso máximo sendo 10 segundos, enquanto em N_2 , 99% dos pacotes experimentam menos de um segundo de atraso, sem limite de atraso máximo. Discuss como o RTP pode ser usado nesses dois casos para transmitir stream de áudio / vídeo ao vivo.

22. Qual é o tamanho total do TCP MTU mínimo, incluindo TCP e sobrecarga de IP, mas não incluindo sobrecarga da camada de enlace?

23. A fragmentação e remontagem do datagrama são tratadas pelo IP e são invisíveis para o TCP. Isso significa que o TCP não precisa se preocupar com os dados chegando de forma errada ordem?

24. RTP é usado para transmitir áudio com qualidade de CD, o que faz um par de amostras de 16 bits 44.100 vezes / seg, uma amostra para cada um dos canais estéreo. Quantos pacotes por em segundo lugar, o RTP deve transmitir?

25. Seria possível colocar o código RTP no kernel do sistema operacional, junto com o código UDP? Explique sua resposta.

26. Um processo no host 1 foi atribuído à porta p , e um processo no host 2 foi porta atribuída q . É possível haver duas ou mais conexões TCP entre essas duas portas ao mesmo tempo?

27. Na Figura 6-36, vimos que, além do campo de *reconhecimento* de 32 bits, há um ACK bit na quarta palavra. Isso realmente acrescenta alguma coisa? Por que ou por que não?

28. A carga útil máxima de um segmento TCP é de 65.495 bytes. Por que foi tão estranho número escolhido?

Página 633

INDIVÍDUO. 6 PROBLEMAS

609

29. Descreva duas maneiras de entrar no estado *SYN RCVD* da Figura 6.39.

30. Considere o efeito de usar o início lento em uma linha com um tempo de ida e volta de 10 ms e nenhum congestionamento. A janela de recepção é de 24 KB e o tamanho máximo do segmento é de 2 KB. Quanto tempo leva para que a primeira janela completa possa ser enviada?

31. Suponha que a janela de congestionamento TCP esteja definida para 18 KB e ocorra um tempo limite. Como grande será a janela se as próximas quatro rajadas de transmissão forem todas bem-sucedidas? Presumir que o tamanho máximo do segmento é 1 KB.

32. Se o tempo de ida e volta do TCP, RTT , é atualmente 30 mseg e o seguinte confirmamentos vêm após 26, 32 e 24 ms, respectivamente, qual é a nova estimativa de RTT usando o algoritmo de Jacobson? Use $\alpha = 0,9$.

33. Uma máquina TCP está enviando janelas completas de 65.535 bytes em um canal de 1 Gbps que tem um atraso unidirecional de 10 ms. Qual é a taxa de transferência máxima alcançável? O que é a eficiência da linha?

34. Qual é a velocidade de linha mais rápida em que um host pode enviar cargas de TCP de 1500 bytes com uma vida útil máxima do pacote de 120 segundos sem que os números de sequência sejam agrupados por aí? Leve a sobrecarga de TCP, IP e Ethernet em consideração. Suponha que Ether-quadros de rede podem ser enviados continuamente.

35. Para resolver as limitações do IP versão 4, um grande esforço teve que ser realizado por meio de IETF que resultou no design da versão 6 do IP e ainda há uma relutância significativa importância na adoção desta nova versão. No entanto, nenhum grande esforço é necessário para abordar as limitações do TCP. Explique por que esse é o caso.

36. Em uma rede cujo segmento máximo é de 128 bytes, a vida útil máxima do segmento é de 30 segundos, e tem números de sequência de 8 bits, qual é a taxa máxima de dados por conexão?

37. Suponha que você esteja medindo o tempo para receber um segmento. Quando uma interrupção ocorre, você lê o relógio do sistema em milissegundos. Quando o segmento é totalmente prôcessado, você leu o relógio novamente. Você mede 0 mseg 270.000 vezes e 1 mseg 730.000 vezes. Quantos tempo leva para receber um segmento?

38. Uma CPU executa instruções à taxa de 1000 MIPS. Os dados podem ser copiados de 64 bits em um tempo, com cada palavra copiada custando 10 instruções. Se um pacote que vem tem que ser copiado quatro vezes, este sistema pode lidar com uma linha de 1 Gbps? Para simplificar, suponha que todas as instruções, mesmo as instruções que lêem ou gravam na memória, são executadas por completo. Taxa de 1000 MIPS.

39. Para contornar o problema de números de sequência envolvendo pacotes antigos

ainda existem, pode-se usar números de sequência de 64 bits. No entanto, teoricamente, uma óptica a fibra pode operar a 75 Tbps. Qual a vida útil máxima do pacote é necessária para garantir que futuras redes de 75 Tbps não têm problemas de wraparound, mesmo com a sequência de 64 bits números? Suponha que cada byte tenha seu próprio número de sequência, como o TCP.

40. Na seção 6.6.5, calculamos que uma linha de gigabit despeja 80.000 pacotes / s no host, dando-lhe apenas 6250 instruções para processá-lo e deixando metade do tempo da CPU para o aplicativo cátions. Este cálculo assumiu um pacote de 1500 bytes. Refaça o cálculo para um Pacote de tamanho ARPANET (128 bytes). Em ambos os casos, suponha que os tamanhos dos pacotes dados incluem todas as despesas gerais.

Página 634

610

A CAMADA DE TRANSPORTE

INDIVÍDUO. 6

41. Para uma rede de 1 Gbps operando acima de 4000 km, o atraso é o fator limitante, não o largura de banda. Considere um MAN com origem e destino médios de 20 km de distância. Em qual taxa de dados o atraso de ida e volta devido à velocidade da luz é igual à transmissão sion delay para um pacote de 1 KB?

42. Calcule o produto de atraso de largura de banda para as seguintes redes: (1) T1 (1,5 Mbps), (2) Ethernet (10 Mbps), (3) T3 (45 Mbps) e (4) STS-3 (155 Mbps). Suponha que um RTT de 100 mseg. Lembre-se de que um cabeçalho TCP tem 16 bits reservados para o tamanho da janela. Quais são suas implicações à luz de seus cálculos?

43. Qual é o produto de atraso de largura de banda para um canal de 50 Mbps em um satélite geoestacionário Leve? Se os pacotes são todos de 1500 bytes (incluindo overhead), quão grande deve a vitória dow estar em pacotes?

44. O servidor de arquivos da Figura 6.6 está longe de ser perfeito e poderia ser melhorado. Faça as seguintes modificações.

(a) Dê ao cliente um terceiro argumento que especifica um intervalo de bytes.

(b) Adicione um sinalizador de cliente -w que permite que o arquivo seja gravado no servidor.

45. Uma função comum que todos os protocolos de rede precisam é manipular mensagens. Lembre-se de que os protocolos manipulam mensagens adicionando / separando cabeçalhos. Alguns protocolos pode quebrar uma única mensagem em vários fragmentos e, posteriormente, juntar esses vários fragmentos mentos de volta em uma única mensagem. Para este fim, projete e implemente uma mensagem biblioteca de gerenciamento que fornece suporte para a criação de uma nova mensagem, anexando um cabeçalho para uma mensagem, retirando um cabeçalho de uma mensagem, dividindo uma mensagem em duas mensagens, combinando duas mensagens em uma única mensagem e salvando uma cópia de uma mensagem sábio. Sua implementação deve minimizar a cópia de dados de um buffer para outro como tanto quanto possível. É fundamental que as operações que manipulam mensagens não toque nos dados em uma mensagem, mas, em vez disso, manipule apenas ponteiros.

46. Projete e implemente um sistema de bate-papo que permita que vários grupos de usuários conversem. UMA o coordenador do chat reside em um endereço de rede conhecido, usa UDP para comunicação com clientes de chat, configura servidores de chat para cada sessão de chat e mantém um chat diretório de sessão. Existe um servidor de chat por sessão de chat. Um servidor de bate-papo usa TCP para comunicação com clientes. Um cliente de bate-papo permite aos usuários iniciar, entrar e sair de um sessão de chat. Projete e implemente o coordenador, o servidor e o código do cliente.

Página 635

7

A CAMADA DE APLICAÇÃO

Tendo terminado todas as preliminares, chegamos agora à camada onde todos os aplicativos são encontrados. As camadas abaixo da camada de aplicação existem para fornecer serviços de transporte, mas eles não fazem um trabalho real para os usuários. Neste capítulo, iremos estudar algumas aplicações de rede reais.

No entanto, mesmo na camada de aplicação, há necessidade de protocolos de suporte, para permitir que os aplicativos funcionem. Assim, veremos um importante

antes de começar com os próprios aplicativos. O item em questão é DNS, que lida com a nomenclatura na Internet. Depois disso, vamos examinar três aplicações reais: correio eletrônico, World Wide Web e multimídia. Terminaremos o capítulo dizendo mais sobre distribuição de conteúdo, inclusive por redes ponto a ponto.

7.1 DNS - O SISTEMA DE NOME DE DOMÍNIO

Embora os programas teoricamente possam se referir a páginas da Web, caixas de correio e outros recursos usando os endereços de rede (por exemplo, IP) dos computadores em quais são armazenados, esses endereços são difíceis de lembrar. Além disso, navegar nas páginas da Web de uma empresa de *128.111.24.41* significa que se a empresa move o servidor da Web para uma máquina diferente com um endereço IP diferente, todos precisa ser informado do novo endereço IP. Consequentemente, nomes legíveis de alto nível foram introduzidos para separar os nomes das máquinas dos endereços das máquinas. No

611

Página 636

612

A CAMADA DE APLICAÇÃO
INDIVÍDUO. 7

dessa forma, o servidor da Web da empresa pode ser conhecido como www.cs.washington.edu independentemente de seu endereço IP. No entanto, uma vez que a própria rede entende apenas endereços numéricos, algum mecanismo é necessário para converter os nomes para endereços de rede. Nas seções a seguir, estudaremos como esse mapeamento é realizado na Internet.

Nos tempos da ARPANET, havia simplesmente um arquivo, *hosts.txt*, que listava todos os nomes de computador e seus endereços IP. Todas as noites, todos os anfitriões iriam busque-o no site em que foi mantido. Para uma rede de algumas centenas grandes máquinas de compartilhamento de tempo, essa abordagem funcionou razoavelmente bem. No entanto, muito antes de muitos milhões de PCs serem conectados à Internet, todos os envolvidos perceberam que essa abordagem não poderia continuar a funcionar para sempre. Por um lado, o tamanho do arquivo ficaria muito grande. Contudo, ainda mais importante, conflitos de nome de host ocorreriam constantemente, a menos que os nomes eram gerenciados centralmente, algo impensável em uma enorme rede internacional devido à carga e latência. Para resolver esses problemas, **DNS (Domain Name System)** foi inventado em 1983. Desde então, tem sido uma parte fundamental da Internet. A essência do DNS é a invenção de uma nomenclatura hierárquica baseada em domínio esquema e um sistema de banco de dados distribuído para implementar este esquema de nomenclatura.

É usado principalmente para mapear nomes de host para endereços IP, mas também pode ser usado para outros fins. DNS é definido em RFCs 1034, 1035, 2181 e mais elaborado em muitos outros.

Resumidamente, a forma como o DNS é usado é a seguinte. Para mapear um nome em um IP endereço, um programa de aplicação chama um procedimento de biblioteca chamado **resolvedor**, pass-

cante o nome como parâmetro. Vimos um exemplo de resolvedor, *gethostbyname*, na Figura 6-6. O resolvedor envia uma consulta contendo o nome para um local Servidor DNS, que procura o nome e retorna uma resposta contendo o anúncio de IP vestido para o resolvedor, que o retorna para o chamador. A consulta e resposta as mensagens são enviadas como pacotes UDP. Armado com o endereço IP, o programa pode em seguida, estabeleça uma conexão TCP com o host ou envie pacotes UDP.

7.1.1 O Espaço de Nome DNS

Gerenciar um grande conjunto de nomes em constante mudança é um problema não trivial. No sistema postal, o gerenciamento de nomes é feito exigindo que as letras especifiquem *fy* (implícita ou explicitamente) o país, estado ou província, cidade, endereço e nome do destinatário. Usar este tipo de endereçamento hierárquico garante que não há confusão entre o Marvin Anderson na Main St. em White Plains, NY e o Marvin Anderson na Main St. em Austin, Texas. DNS funciona

da mesma maneira.

Para a Internet, o topo da hierarquia de nomenclatura é gerenciado por uma organização chamada **ICANN** (**Internet Corporation for Assigned Names and Numbers**).

ICANN foi criada para esse fim em 1998, como parte do amadurecimento do Inter rede para uma preocupação econômica mundial. Conceitualmente, a Internet é dividida em

Página 637

SEC. 7,1

DNS - O SISTEMA DE NOME DE DOMÍNIO

613

mais de 250 **domínios de nível superior**, onde cada domínio cobre muitos hosts. Cada um main é particionado em subdomínios e estes são particionados posteriormente e assim por diante. Todos esses domínios podem ser representados por uma árvore, conforme mostrado na Figura 7-1. As folhas

da árvore representam domínios que não têm subdomínios (mas contêm máquinas, claro). Um domínio folha pode conter um único host, ou pode representar um componente e conter milhares de hosts.

```
...
eng
cisco
ieee
acm
eng
Washington
cs
robô
Jill
macaco
co
ac
csl
nec
cs
Keio
uwa
edu
oce
vu
lei
cs
edu
museu
aero
com
gov
org
jp
nos
internet
au
Reino Unido
nl
Genérico
Países
...
flauta
Filt
```

Figura 7-1. Uma parte do espaço de nomes de domínio da Internet.

Os domínios de nível superior vêm em dois sabores: genérico e países. O gen- domínios eric, listados na Fig. 7-2, incluem domínios originais da década de 1980 e principais introduzidos por meio de solicitações à ICANN. Outros domínios genéricos de nível superior

será adicionado no futuro.

Os domínios de país incluem uma entrada para cada país, conforme definido na ISO 3166. Nomes de domínio de países internacionalizados que usam alfabetos não latinos eram introduzido em 2010. Esses domínios permitem que as pessoas nomeiem hosts em árabe, cirílico, Chinês ou outros idiomas.

Obter um domínio de segundo nível, como *name-of-company.com*, é fácil. o

Os domínios de primeiro nível são administrados por **registradores** indicados pela ICANN. Obtendo um nome

requer apenas ir a um registrador correspondente (para *com*, neste caso) para verificar se o nome desejado estiver disponível e não a marca registrada de outra pessoa. Se houver sem problemas, o solicitante paga ao registrador uma pequena taxa anual e obtém o nome. No entanto, à medida que a Internet se tornou mais comercial e mais internacional,

al, também se tornou mais contencioso, especialmente em questões relacionadas à nomenclatura. Essa controvérsia inclui a própria ICANN. Por exemplo, a criação do xxx domain levou vários anos e processos judiciais para serem resolvidos. Está colocando voluntariamente um adulto

o conteúdo em seu próprio domínio é bom ou ruim? (Algumas pessoas não queriam adulto conteúdo disponível na Internet, enquanto outros queriam colocar tudo em um só fazer principal, para que os filtros de babá possam facilmente localizar e bloquear o acesso de crianças). Alguns dos

domínios se auto-organizam, enquanto outros têm restrições sobre quem pode obter um nome, conforme observado na Fig. 7-2. Mas quais restrições são apropriadas? Pegue o domínio *profissional*,

Página 638

614

A CAMADA DE APLICAÇÃO INDIVÍDUO. 7

Domínio

Uso pretendido

Data de início restrita?

com

Comercial

1985

Não

edu

Instituições educacionais

1985

sim

gov

Governo

1985

sim

int

Organizações internacionais

1988

sim

mil

Militares

1985

sim

internet

Provedores de rede

1985

Não

org

Organizações sem fins lucrativos

1985

Não

aero

Transporte aéreo

2001

sim

negócios

Negócios

2001

Não

coop

Cooperativas

2001

sim

informação

Informativo

2002

Não

museu

Museus

2002

sim

nome	
Pessoas	
2002	
Não	
pró	
Profissionais	
2002	
sim	
gato	
catalão	
2005	
sim	
empregos	
Emprego	
2005	
sim	
mobi	
Dispositivos móveis	
2005	
sim	
tel	
Detalhes do contato	
2005	
sim	
viagem	
Indústria de viagens	
2005	
sim	
xxx	
Indústria do sexo	
2010	
Não	

Figura 7-2. Domínios genéricos de nível superior.

por exemplo. É para profissionais qualificados. Mas quem é profissional? Doutores e os advogados são claramente profissionais. Mas e quanto aos fotógrafos freelance, professores de piano, mágicos, encanadores, barbeiros, exterminadores, tatuadores, mercenários e prostitutas? Essas ocupações são elegíveis? De acordo com quem?

Também há dinheiro nos nomes. Tuvalu (o país) vendeu um aluguel de sua *tv* principal por US \$ 50 milhões, tudo porque o código do país é adequado para publicidade sites de televisão. Praticamente todas as palavras comuns (em inglês) foram tomadas no domínio *com*, junto com os erros ortográficos mais comuns. Experimente artigos domésticos, animais, plantas, partes do corpo, etc. A prática de registrar um domínio apenas para virar por aí e vendê-lo para uma parte interessada a um preço muito mais alto, mesmo que tenha um nome. É chamado de **cybersquatting**. Muitas empresas que estavam atrasadas quando a era da Internet começou, seus nomes de domínio óbvios já estavam em uso quando eles tentaram adquiri-los. Em geral, enquanto nenhuma marca registrada estiver sendo violado e sem fraude envolvida, é o primeiro a chegar, o primeiro a ser servido com os nomes. Nunca-

no entanto, as políticas para resolver disputas de nomenclatura ainda estão sendo refinadas.

Página 639

SEC. 7,1

DNS - O SISTEMA DE NOME DE DOMÍNIO

615

Cada domínio é nomeado pelo caminho ascendente a partir dele até a raiz (sem nome). os componentes são separados por pontos (pronuncia-se " ponto "). Assim, a engenharia departamento da Cisco pode ser *eng.cisco.com.* , em vez de um nome no estilo UNIX, como *as / com / cisco / eng*. Observe que essa nomenclatura hierárquica significa que *eng.cisco.com.* não entra em conflito com o uso potencial de *eng* em *eng.washington.edu.* , que poderá ser usado pelo departamento de inglês da Universidade de Washington.

Os nomes de domínio podem ser absolutos ou relativos. Um nome de domínio absoluto sempre termina com um ponto (por exemplo, *eng.cisco.com.*), enquanto um parente não.

Os nomes relativos devem ser interpretados em algum contexto para determinar de forma exclusiva seus verdadeiros significados. Em ambos os casos, um domínio nomeado se refere a um nó específico na árvore e todos os nós abaixo dele.

Os nomes de domínio não diferenciam maiúsculas de minúsculas, então *edu*, *Edu* e *EDU* significam o mesmo coisa. Os nomes dos componentes podem ter até 63 caracteres e os nomes dos caminhos completos não deve exceder 255 caracteres.

Em princípio, os domínios podem ser inseridos na árvore em genéricos ou de país domínios. Por exemplo, *cs.washington.edu* também poderia ser listado no *nos* domínio do país como *cs.washington.wa.us*. Na prática, no entanto, a maioria das organizações estão nos Estados Unidos sob domínios genéricos e a maioria fora dos Estados Unidos sob o domínio de seu país. Não há regra contra o registro em vários domínios de nível superior. Grandes empresas costumam fazer isso (por exemplo, *sony.com*, *sony.net* e *sony.nl*).

Cada domínio controla como aloca os domínios sob ele. Por exemplo, O Japão tem os domínios *ac.jp* e *co.jp* que espelham *edu* e *com*. A Holanda faz não faz essa distinção e coloca todas as organizações diretamente sob a *nl*. Assim, todos três dos seguintes são departamentos universitários de ciência da computação:

1. *cs.washington.edu* (Universidade de Washington, nos EUA).
2. *cs.vu.nl* (Vrije Universiteit, na Holanda).
3. *cs.keio.ac.jp* (Keio University, no Japão).

Para criar um novo domínio, é necessária a permissão do domínio no qual será ser incluído. Por exemplo, se um grupo VLSI for iniciado na University of Washington e quer ser conhecido como *vlsi.cs.washington.edu*, precisa de permissão de quem gerencia *cs.washington.edu*. Da mesma forma, se uma nova universidade é chamada, digamos, da University of Northern South Dakota, deve perguntar ao gerente de o domínio *edu* para atribuí-lo a *unsd.edu* (se ainda estiver disponível). Desta forma, nome conflitos são evitados e cada domínio pode controlar todos os seus subdomínios. Uma vez um novo domínio foi criado e registrado, ele pode criar subdomínios, como *cs.unsd.edu*, sem obter permissão de ninguém no topo da árvore.

A nomenclatura segue limites organizacionais, não redes físicas. Para exemplo, se os departamentos de ciência da computação e engenharia elétrica estiverem localizados no mesmo prédio e compartilhar a mesma LAN, eles podem, no entanto, ter

Página 640

616

A CAMADA DE APLICAÇÃO INDIVÍDUO. 7

domínios. Da mesma forma, mesmo se a ciência da computação estiver dividida entre Babbage Hall e Turing Hall, os hosts em ambos os edifícios pertencerão normalmente ao mesmo domínio.

7.1.2 Registros de Recursos de Domínio

Cada domínio, seja um único host ou um domínio de nível superior, pode ter um conjunto de **registros de recursos** associados a ele. Esses registros são o banco de dados DNS. Para um único host, o registro de recurso mais comum é apenas seu endereço IP, mas muitos outros tipos de registros de recursos também existem. Quando um resolvelor fornece um nome de domínio

ao DNS, o que ele recebe são os registros de recursos associados a esse nome. Assim, a função principal do DNS é mapear nomes de domínio em recursos registros.

Um registro de recurso é uma tupla de cinco. Embora sejam codificados em binário para eficiência, na maioria dos registros de recursos de exposições são apresentados como texto ASCII, um

linha por registro de recurso. O formato que usaremos é o seguinte:

Nome de domínio Tempo de vida Tipo de classe Valor

O *nome* do domínio informa o domínio ao qual este registro se aplica. Normalmente, muitos existem registros para cada domínio e cada cópia do banco de dados contém informações sobre vários domínios. Este campo é, portanto, a chave de pesquisa primária usada para satisfazer consultas. A ordem dos registros no banco de dados não é significativa.

O campo *Time to live* dá uma indicação de quanto estável é o registro. Informação que é altamente estável é atribuído um grande valor, como 86400 (o número de segundos em 1 dia). As informações altamente voláteis recebem um pequeno valor, como 60 (1 minuto). Voltaremos a este ponto mais tarde, quando tivermos discussões malditas.

O terceiro campo de cada registro de recurso é a *Classe*. Para informações da Internet, está sempre *DENTRO*. Para informações que não sejam da Internet, outros códigos podem ser usados, mas em prática estes raramente são vistos.

O campo *Tipo* informa que tipo de registro é este. Existem muitos tipos de DNS registros. Os tipos importantes estão listados na Figura 7-3.

Um registro *SOA* fornece o nome da fonte primária de informações sobre a zona do servidor de nomes (descrita abaixo), o endereço de e-mail de seu administrador, um número de série exclusivo e vários sinalizadores e tempos limite.

O tipo de registro mais importante é o registro *A* (Endereço). Ele contém um 32-bit Endereço IPv4 de uma interface para algum host. O *AAAA* correspondente, ou "quad

Um registro " contém um endereço IPv6 de 128 bits. Cada host da Internet deve ter pelo menos um endereço IP para que outras máquinas possam se comunicar com ele. Alguns anfitriões têm duas ou mais interfaces de rede, caso em que terão duas ou mais interfaces de tipo *A* ou *registros de recursos AAAA*. Consequentemente, o DNS pode retornar vários endereços para um único nome.

Um tipo de registro comum é o registro *MX*. Ele especifica o nome do host preparado para aceitar e-mail para o domínio especificado. É usado porque nem todo

Página 641

SEC. 7,1

DNS - O SISTEMA DE NOME DE DOMÍNIO

617

Tipo

Significado

Valor

SOA

Início de autoridade

Parâmetros para esta zona

UMA

Endereço IPv4 de um host

Inteiro de 32 bits

AAAA

Endereço IPv6 de um host

Inteiro de 128 bits

MX

Troca de correio

Prioridade, domínio disposto a aceitar e-mail

NS

Nome do servidor

Nome de um servidor para este domínio

CNAME

Nome canônico

Nome do domínio

PTR

Pointer

Alias para um endereço IP

SPF

Estrutura da política do remetente

Codificação de texto da política de envio de e-mail

SRV

Serviço

Host que o fornece

TXT
Texto
Texto ASCII descritivo

Figura 7-3. Os principais tipos de registro de recurso DNS.

máquina está preparada para aceitar e-mail. Se alguém quiser enviar e-mail para, por exemplo, *bill@microsoft.com*, o host de envio precisa encontrar algum servidor de correio local *ted* em *microsoft.com* que está disposto a aceitar e-mail. O registro *MX* pode fornecer essa informação.

Outro tipo de registro importante é o registro *NS*. Ele especifica um servidor de nomes para o domínio ou subdomínio. Este é um host que possui uma cópia do banco de dados para um a Principal. É usado como parte do processo de pesquisa de nomes, que iremos descrever Em breve.

Os registros *CNAME* permitem a criação de aliases. Por exemplo, uma pessoa familiar com nomes de Internet em geral e querendo mandar uma mensagem para o usuário *paul* no departamento de ciência da computação do MIT pode adivinhar que *paul@cs.mit.edu* vai trabalhos. Na verdade, este endereço não funcionará, porque o domínio para comp do MIT o departamento de ciências do computador é *csail.mit.edu*. No entanto, como um serviço para pessoas que fazem não sei disso, o MIT poderia criar uma entrada *CNAME* para apontar pessoas e programas na direção certa. Uma entrada como esta pode fazer o trabalho:

cs.mit.edu 86400 IN CNAME csail.mit.edu

Como *CNAME*, *PTR* aponta para outro nome. No entanto, ao contrário do *CNAME*, que é realmente apenas uma definição de macro (ou seja, um mecanismo para substituir uma string por outra

er), *PTR* é um tipo de dados DNS regular, cuja interpretação depende do contexto em que se encontra. Na prática, quase sempre é usado para associar um nome a um endereço IP para permitir pesquisas do endereço IP e retornar o nome do correspondente máquina de ponderação. Eles são chamados **de pesquisas reversas**.

SRV é um novo tipo de registro que permite que um host seja identificado para um determinado serviço em um domínio. Por exemplo, o servidor da Web para *cs.washington.edu* pode ser identificado como *cockatoo.cs.washington.edu*. Este registro generaliza o registro *MX* que executa a mesma tarefa, mas é apenas para servidores de e-mail.

Página 642

618

A CAMADA DE APLICAÇÃO INDIVÍDUO. 7

SPF também é um novo tipo de registro. Ele permite que um domínio codifique informações sobre quais máquinas no domínio enviarão correio para o resto da Internet. Isso ajuda máquinas receptoras verificam se o e-mail é válido. Se o correio estiver sendo recebido de um ma que se autodenomina *duvidosa*, mas os registros de domínio dizem que o e-mail só será enviado fora do domínio por uma máquina chamada *smtp*, é provável que o e-mail seja forjado lixo eletrônico.

Por último na lista, os registros *TXT* foram fornecidos originalmente para permitir que os domínios se identificam de maneiras arbitrárias. Hoje em dia, eles geralmente codificam máquinas informações legíveis, normalmente as informações *SPF*.

Finalmente, temos o campo *Valor*. Este campo pode ser um número, um nome de domínio, ou uma string ASCII. A semântica depende do tipo de registro. Uma breve descrição dos campos *Value* para cada um dos principais tipos de registro é apresentado na Figura 7.3.

Para obter um exemplo do tipo de informação que pode ser encontrada no banco de dados DNS de um domínio, consulte a Fig. 7-4. Esta figura representa parte de um banco de dados (hipotético) para

o domínio *cs.vu.nl* mostrado na Fig. 7-1. O banco de dados contém sete tipos de re registrados de origem.

; Dados oficiais para cs.vu.nl

cs.vu.nl

86400 IN SOA

chefe estrela (9527,7200,7200,241920,86400)

cs.vu.nl.

```

86400 IN MX
1 zéfiro
cs.vu.nl.
86400 IN MX
2 principais
cs.vu.nl.
86400 IN NS
Estrela
Estrela
86400 IN A
130.37.56.205
zéfiro
86400 IN A
130.37.20.10
topo
86400 IN A
130.37.20.11
www
86400 IN CNAME
star.cs.vu.nl
ftp
86400 IN CNAME
zephyr.cs.vu.nl
esvoaçantes
86400 IN A
130.37.16.112
esvoaçantes
86400 IN A
192.31.231.165
esvoaçantes
86400 IN MX
1 flits
esvoaçantes
86400 IN MX
2 zéfiro
esvoaçantes
86400 IN MX
3 melhores
barco a remo
EM UM
130.37.56.201
IN MX
1 barco a remo
IN MX
2 zéfiro
irmãzinha
EM UM
130.37.62.23
laserjet
EM UM
192.31.231.216

```

Figura 7-4. Uma porção de uma base de dados possível de DNS para *cs.vu.nl* .

A primeira linha sem comentários da Figura 7-4 fornece algumas informações básicas sobre o domínio, que não nos preocupará mais. Em seguida, vêm duas entradas, dando a primeira

Página 643

SEC. 7,1
DNS - O SISTEMA DE NOME DE DOMÍNIO

619

e segundos lugares para tentar entregar e-mail enviado para *person@cs.vu.nl* . O *zéfiro* (a máquina específica) deve ser tentado primeiro. Se isso falhar, o *topo* deve ser tentado como o próxima escolha. A próxima linha identifica o servidor de nomes para o domínio como *estrela* . Após a linha em branco (adicionada para facilitar a leitura), vêm as linhas que fornecem os endereços IP para a *estrela* , *zéfiro* e *topo* . Eles são seguidos por um alias, www.cs.vu.nl , para que este endereço pode ser usado sem designar uma máquina específica. Criando este alias permite que *cs.vu.nl* altere seu servidor da World Wide Web sem invalidar

o endereço que as pessoas usam para chegar até ele. Um argumento semelhante é válido para *ftp.cs.vu.nl* .

A seção para os *flits* da máquina lista dois endereços IP e três opções são fornecido para lidar com o e-mail enviado para *flits.cs.vu.nl* . A primeira escolha é, naturalmente, o *flits* it-self, mas se for baixo, o *zéfiro* e o *topo* são a segunda e a terceira opções.

As próximas três linhas contêm uma entrada típica para um computador, neste caso, *rowboat.cs.vu.nl* . As informações fornecidas contêm o endereço IP e o pri-Mary e correio secundário cai. Em seguida, vem uma entrada para um computador que não é capaz de receber a própria correspondência, seguida por uma entrada que é provavelmente para uma impressora que está conectado à Internet.

7.1.3 Servidores de Nomes

Em teoria, pelo menos, um único servidor de nomes pode conter todo o banco de dados DNS e responder a todas as dúvidas sobre o assunto. Na prática, este servidor estaria tão sobrecarregado quanto a ser inútil. Além disso, se alguma vez caísse, toda a Internet seria aleijado.

Para evitar os problemas associados a ter apenas uma única fonte de informações mação, o espaço de nomes DNS é dividido em **zonas** não sobrepostas . Um possível A maneira de dividir o espaço de nomes da Fig. 7-1 é mostrada na Fig. 7-5. Cada zona circulada contém alguma parte da árvore.

```
...  
eng  
cisco  
ieee  
acm  
eng  
Washington  
cs  
robô  
Jill  
macaco  
co  
ac  
csl  
nec  
cs  
Keio  
uwa  
edu  
oce  
vu  
lei  
cs  
edu  
museu  
aero  
com  
gov  
org  
jp  
nos  
internet  
au  
Reino Unido  
nl  
Genérico  
Países  
...  
flauta  
esvoaçantes
```

Figura 7-5. Parte do espaço de nomes DNS dividido em zonas (que estão circuladas).

Página 644

620

A CAMADA DE APLICAÇÃO INDIVÍDUO. 7

Onde os limites da zona são colocados dentro de uma zona é até o anúncio dessa zona ministrador. Esta decisão é feita em grande parte com base em quantos servidores de nomes são desejados e onde. Por exemplo, na Figura 7-5, a Universidade de Washington tem uma zona para *washington.edu* que manipula *eng.washington.edu*, mas não *hangle.cs.washington.edu* . Essa é uma zona separada com seus próprios servidores de nomes. Tal decisão pode ser tomada quando um departamento como o Inglês não deseja executar seu próprio servidor de nomes, mas um departamento como Ciência da Computação tem.

Cada zona também está associada a um ou mais servidores de nomes. Estes são hospedeiros que contém o banco de dados da zona. Normalmente, uma zona terá um nome principal servidor, que obtém suas informações de um arquivo em seu disco, e um ou mais segundos servidores de nomes secundários, que obtêm suas informações do servidor de nomes primário. Para melhorar a confiabilidade, alguns dos servidores de nomes podem estar localizados fora da zona.

O processo de procurar um nome e encontrar um endereço é chamado de **nome resolution**. Quando um resolvelor tem uma consulta sobre um nome de domínio, ele passa a consulta para um servidor de nomes local. Se o domínio procurado estiver sob a jurisdição do servidor de nomes, como *top.cs.vu.nl* caindo em *cs.vu.nl*, ele retorna o autoritativo registros de recursos. Um **registro oficial** é aquele que vem da autoridade que gerencia o registro e, portanto, está sempre correto. Os registros oficiais estão em contraste com os **registros em cache**, que podem estar desatualizados. O que acontece quando o domínio é remoto, como quando *flits.cs.vu.nl* deseja encontrar o endereço IP de *robot.cs.washington.edu* em UW (University of Washington-telada)? Neste caso, e se não houver informações em cache sobre a disponibilidade do domínio localmente, o servidor de nomes inicia uma consulta remota. Esta consulta segue o processo mostrado na Fig. 7-6. A etapa 1 mostra a consulta que é enviada ao serviço de nome local ver. A consulta contém o nome de domínio procurado, o tipo (*A*) e a classe (*IN*).

```
10: robot.cs.washington.edu
1: consulta
2: consulta
3: edu
5: washington.edu
4: consulta
6: consulta
7: cs.washington.edu
9: robot.cs.washington.edu
8: consulta
Local
(cs.vu.nl)
nome do servidor
UWCS
nome do servidor
UW
nome do servidor
Servidor de nomes edu
(a.edu-servers.net)
Servidor de nome raiz
(a.root-servers.net)
flits.cs.vu.nl
Origínador
```

Figura 7-6. Exemplo de um resolvelor procurando um nome remoto em 10 etapas.

A próxima etapa é começar no topo da hierarquia de nomes, perguntando a um dos **servidores de nomes raiz**. Esses servidores de nomes têm informações sobre cada nível superior

Página 645

SEC. 7,1

DNS - O SISTEMA DE NOME DE DOMÍNIO

621

domínio. Isso é mostrado como etapa 2 na Fig. 7-6. Para entrar em contato com um servidor raiz, cada nome

O servidor deve ter informações sobre um ou mais servidores de nomes raiz. Este informação está normalmente presente em um arquivo de configuração do sistema que é carregado no

Cache DNS quando o servidor DNS é iniciado. É simplesmente uma lista de registros *NS* para a raiz e os registros *A* correspondentes.

Existem 13 servidores DNS raiz, chamados sem imaginação de *a-root-servers.net* por meio de *m.root-servers.net*. Cada servidor raiz pode ser logicamente um único computador. No entanto, como toda a Internet depende dos servidores raiz, eles são poderosos e computadores altamente replicados. A maioria dos servidores estão presentes em vários localizações gráficas e alcançadas usando o roteamento anycast, no qual um pacote é entregue remetido para a instância mais próxima de um endereço de destino; nós descrevemos anycast em Indivíduo. 5 A replicação melhora a confiabilidade e o desempenho.

É improvável que o servidor de nomes raiz saiba o endereço de uma máquina em UW, e provavelmente também não conhece o servidor de nomes para UW. Mas deve conhecer o

servidor de nomes para o domínio *edu* , no qual *cs.washington.edu* está localizado. Retorna o nome e o endereço IP para essa parte da resposta na etapa 3.

O servidor de nomes local então continua sua busca. Ele envia toda a consulta para o servidor de nomes *edu* (*a.edu-servers.net*). Esse servidor de nome retorna o servidor de nome para UW. Isso é mostrado nas etapas 4 e 5. Mais perto agora, o servidor de nomes local envia a consulta ao servidor de nomes UW (etapa 6). Se o nome de domínio procurado fosse no departamento de inglês, a resposta seria encontrada, pois a zona UW inclui o departamento de inglês. Mas o departamento de Ciência da Computação optou por administrar seu próprio servidor de nomes. A consulta retorna o nome e o endereço IP do UW Com-servidor de nomes da puter Science (etapa 7).

Por fim, o servidor de nomes local consulta o servidor de nomes UW Computer Science (etapa 8). Este servidor é autorizado para o domínio *cs.washington.edu* , então ele deve tem a resposta. Ele retorna a resposta final (etapa 9), que o servidor de nomes local encaminha como uma resposta a *flits.cs.vu.nl* (etapa 10). O nome foi resolvido.

Você pode explorar este processo usando ferramentas padrão, como o programa *dig* que está instalado na maioria dos sistemas UNIX . Por exemplo, digitando
dig@a.edu-servers.net robot.cs.washington.edu
irá enviar uma consulta para *robot.cs.washington.edu* para o nome ser- *a.edu-servers.net* ver e imprimir o resultado. Isso mostrará as informações obtidas na etapa 4 no exemplo acima, e você aprenderá o nome e o endereço IP do UW servidores de nomes.

Há três pontos técnicos a serem discutidos sobre esse longo cenário. Os dois primeiros diferentes mecanismos de consulta estão em funcionamento na Figura 7.6. Quando o host *flits.cs.vu.nl* envia sua consulta para o servidor de nomes local, esse servidor de nomes trata da resolução em nome da *flits* até que tenha a resposta desejada para retornar. Ele *não* devolução parcial respostas. Eles podem ser úteis, mas não são o que a consulta buscava. Isto mecanismo é chamado de **consulta recursiva** .

Página 646

622

A CAMADA DE APLICAÇÃO

INDIVÍDUO. 7

Por outro lado, o servidor de nomes raiz (e cada servidor de nomes subsequente) não continua recursivamente a consulta para o servidor de nomes local. Ele apenas retorna um resposta parcial e passa para a próxima pergunta. O servidor de nomes local é responsável para continuar a resolução emitindo mais perguntas. Este mecanismo é chamada de **consulta iterativa** .

Uma resolução de nome pode envolver ambos os mecanismos, como este exemplo mostrou.

Uma consulta recursiva pode sempre parecer preferível, mas muitos servidores de nomes (especialmente

(a raiz) não os tratará. Eles estão muito ocupados. As consultas iterativas colocam a carga den no originador. A justificativa para o servidor de nomes local oferecer suporte a um A consulta sive é que ele está fornecendo um serviço aos hosts em seu domínio. Esses anfitriões fazem

não precisa ser configurado para executar um servidor de nomes completo, apenas para acessar o local.

O segundo ponto é o cache. Todas as respostas, incluindo todas as parciais as respostas retornadas são armazenadas em cache. Desta forma, se outro host *cs.vu.nl* consultar *robot.cs.washington.edu* a resposta já será conhecida. Ainda melhor, se um host consultas para um host diferente no mesmo domínio, digamos *galah.cs.washington.edu* , o a consulta pode ser enviada diretamente para o servidor de nomes autorizado. Da mesma forma, as consultas para outros domínios em *washington.edu* podem começar diretamente a partir do nome *washington.edu* servidor. O uso de respostas em cache reduz muito as etapas de uma consulta e melhora desempenho. O cenário original que esboçamos é na verdade o pior caso que ocorreu curs quando nenhuma informação útil é armazenada em cache.

No entanto, as respostas em cache não são autoritativas, uma vez que as alterações feitas em *cs.washington.edu* não serão propagadas para todos os caches no mundo que podem saber sobre isso. Por esse motivo, as entradas de cache não devem durar muito. Isto é o motivo pelo qual o campo *Time to live* está incluído em cada registro de recurso. Diz remotamente aos servidores de nomes quanto tempo para armazenar os registros. Se uma determinada máquina teve o mesmo endereço IP por anos, pode ser seguro armazenar essas informações em cache por 1 dia. Para informações mais voláteis, pode ser mais seguro limpar os registros após alguns segundos ou um minuto.

A terceira questão é o protocolo de transporte que é usado para as consultas e responde. É UDP. As mensagens DNS são enviadas em pacotes UDP com um formato simples para consultas, respostas e servidores de nomes que podem ser usados para continuar a resolução. Não entraremos em detalhes sobre esse formato. Se nenhuma resposta chegar dentro de um curto tempo, o cliente DNS repete a consulta, tentando outro servidor para o domínio após um pequeno número de tentativas. Este processo é projetado para lidar com o caso do servidor estar inativo e também perder o pacote de consulta ou resposta. De 16 bits, o identificador é incluído em cada consulta e copiado para a resposta para que um nome que pode corresponder as respostas à consulta correspondente, mesmo se várias consultas forem executadas ao mesmo tempo.

Mesmo que sua finalidade seja simples, deve ficar claro que o DNS é um grande e complexo sistema distribuído composto por milhões de servidores de nomes que trabalham juntos. Ele forma um elo fundamental entre nomes de domínio legíveis por humanos e os endereços IP das máquinas. Inclui replicação e cache para desempenho e confiabilidade e é projetado para ser altamente robusto.

Página 647

SEC. 7,1

DNS - O SISTEMA DE NOME DE DOMÍNIO

623

Não cobrimos a segurança, mas como você pode imaginar, a capacidade de mudar o mapeamento de nome para endereço pode ter consequências devastadoras se feito maliciosamente. Por esse motivo, extensões de segurança chamadas DNSSEC foram desenvolvidas para DNS. Iremos descrevê-las no cap. 8.

Também há demanda de aplicativos para usar nomes de maneiras mais flexíveis, por exemplo, nomeando o conteúdo e resolvendo para o endereço IP de um host próximo que tem o conteúdo. Isso se encaixa no modelo de busca e download de um filme. Isto é o filme que importa, não o computador que tem uma cópia dele, então tudo o que se deseja é o endereço IP de qualquer computador próximo que tenha uma cópia do filme. Conteúdo nas redes de distribuição são uma forma de realizar esse mapeamento. Vamos descrever como eles se baseiam no DNS posteriormente neste capítulo, na Seç. 7,5.

7.2 CORREIO ELETRÔNICO

O correio eletrônico, ou mais comumente o **e-mail**, existe há mais de três décadas. Mais rápido e mais barato do que o correio em papel, o e-mail tem sido um aplicativo popular

desde os primórdios da Internet. Antes de 1990, era usado principalmente em academia. Durante a década de 1990, tornou-se conhecido do grande público e cresceu exponencialmente, ao ponto em que o número de e-mails enviados por dia agora é muito maior do que o número de **cartas do correio tradicional** (ou seja, de papel). Outras formas de comunicação, como mensagens instantâneas e chamadas de voz sobre IP se expandiram muito em uso na última década, mas o e-mail continua sendo o carro-chefe da Internet comunicação. É amplamente utilizado na indústria para comunicação intracompanhia, por exemplo, para permitir que funcionários distantes em todo o mundo cooperem em projetos complexos. Infelizmente, como a correspondência em papel, a maioria dos e-mails - cerca de 9

em cada 10 mensagens - é lixo eletrônico ou **spam** (McAfee, 2010).

E-mail, como a maioria das outras formas de comunicação, desenvolveu seu próprio con-

teúdo e estilos. É muito informal e tem um baixo limiar de utilização. Pessoas

que nunca sonharia em ligar ou mesmo escrever uma carta para um Muito Importante Pessoa não hesite por um segundo em enviar um e-mail mal escrito para ele ou ela. Ao eliminar a maioria das dicas associadas a classificação, idade e gênero, debates por e-mail muitas vezes se concentram no conteúdo, não no status. Com e-mail, uma ideia brilhante de um estudo de verão um dente pode ter mais impacto do que um idiota de um vice-presidente executivo. Email está cheio de jargões como BTW (By The Way), ROTFL (Rolling On The Rindo do chão), e IMHO (em minha humilde opinião). Muitas pessoas também usam outros símbolos ASCII chamados **smileys**, começando com o onipresente ":-)". Gire o livro 90 graus no sentido horário se este símbolo for desconhecido. Este símbolo e outros **emoticons** ajudam a transmitir o tom da mensagem. Eles se espalharam para outras formas concisas de comunicação, como mensagens instantâneas. Os protocolos de e-mail também evoluíram durante o período de uso. O primeiro sistemas de e-mail consistiam simplesmente em protocolos de transferência de arquivos, com a convenção de que a primeira linha de cada mensagem (ou seja, arquivo) continha o endereço do destinatário. Como tempo

Página 648

624

A CAMADA DE APLICAÇÃO

INDIVÍDUO. 7

continuou, o e-mail divergiu da transferência de arquivos e muitos recursos foram adicionados, como

a capacidade de enviar uma mensagem a uma lista de destinatários. Capacidades de multimídia tornou-se importante na década de 1990 para o envio de mensagens com imagens e outras não textuais

material. Programas para leitura de e-mail também se tornaram muito mais sofisticados, shift-de interfaces de usuário baseadas em texto para gráficas e adicionando a capacidade dos usuários de acessar seus e-mails de seus laptops, onde quer que estejam. Finalmente, com o prevalência de spam, leitores de e-mail e os protocolos de transferência de e-mail agora devem pagar atenção para localizar e remover e-mails indesejados.

Em nossa descrição de e-mail, vamos nos concentrar na maneira como as mensagens de e-mail são movidos entre usuários, em vez da aparência e comportamento dos programas de leitura de e-mail. No entanto, depois de descrever a arquitetura geral, começaremos com o parte do sistema de e-mail voltada para o usuário, já que é familiar para a maioria dos leitores.

7.2.1 Arquitetura e Serviços

Nesta seção, forneceremos uma visão geral de como os sistemas de e-mail são organizados e o que eles podem fazer. A arquitetura do sistema de e-mail é mostrada em

Fig. 7-7. É composto por dois tipos de subsistemas: os **agentes do usuário**, que permitem pessoas para ler e enviar e-mail, e os **agentes de transferência de mensagens**, que movem o mensagens da origem ao destino. Também nos referiremos à mensagem transfer agentes informalmente como **servidores de correio**.

mensagem

Agente transferido

mensagem

Agente transferido

SMTP

Remetente

Agente de usuário

Caixa de correio

Receptor

Agente de usuário

O email

1: Correio

submissão

2: mensagem

transferir

3: final

Entrega

Figura 7-7. Arquitetura do sistema de e-mail.

O agente do usuário é um programa que fornece uma interface gráfica, ou às vezes

uma interface baseada em texto e comando que permite aos usuários interagir com o sistema de e-mail.

Inclui um meio de redigir mensagens e respostas a mensagens, exibir incomensagens e organizá-las arquivando, pesquisando e descartando-as.

O ato de enviar novas mensagens para o sistema de e-mail para entrega é chamado de **e-mail submissão**.

Parte do processamento do agente do usuário pode ser feito automaticamente, antecipando o que o usuário deseja. Por exemplo, o correio recebido pode ser filtrado para extrair ou

Página 649

SEC. 7,2

CORREIO ELETRÔNICO

625

desfaça a prioridade de mensagens que provavelmente sejam spam. Alguns agentes de usuário incluem

recursos, como organizar para respostas automáticas de e-mail ("Estou tendo um férias completas e vai demorar um pouco antes de eu voltar a falar com você"). Um agente de usuário executa

no mesmo computador em que um usuário lê seu e-mail. É apenas mais um programa e pode ser executado apenas algumas vezes.

Os agentes de transferência de mensagens são normalmente processos do sistema. Eles correm no experiência em máquinas de servidor de e-mail e devem estar sempre disponíveis.

O trabalho deles é mover automaticamente o e-mail através do sistema do originador para o destinatário com **SMTP** (**Simple Mail Transfer Protocol**). Esta é a mensagem etapa de transferência.

SMTP foi originalmente especificado como RFC 821 e revisado para se tornar o atual RFC 5321. Ele envia e-mails por meio de conexões e relata o status de entrega e quaisquer erros. Existem numerosos aplicativos em que a confirmação da entrega é importante e pode até ter significado legal ("Bem, Meritíssimo, meu e-mail sistema não é muito confiável, então acho que a intimação eletrônica acabou de se perder algum lugar").

Os agentes de transferência de mensagens também implementam **listas de correio**, nas quais um idêntico

a cópia de uma mensagem é entregue a todos em uma lista de endereços de e-mail. Outro anúncio recursos avançados são cópias carbono, cópias carbono ocultas, e-mail de alta prioridade, segredo (ou seja, criptografado) e-mail, destinatários alternativos se o principal não for atualmente disponível e a capacidade dos assistentes de ler e responder aos e-mails de seus chefes.

Vincular agentes de usuário e agentes de transferência de mensagens são os conceitos de correio caixas e um formato padrão para mensagens de e-mail. **As caixas de correio** armazenam o e-mail que

é recebido para um usuário. Eles são mantidos por servidores de correio. Agentes de usuário simplesmente

apresentar aos usuários uma visão do conteúdo de suas caixas de correio. Para fazer isso, o usuário os agentes enviam os comandos dos servidores de correio para manipular as caixas de correio, inspecionando

seu conteúdo, excluindo mensagens e assim por diante. A recuperação do correio é a de- final pintura (etapa 3) na Fig. 7-7. Com esta arquitetura, um usuário pode usar um usuário diferente agentes em vários computadores para acessar uma caixa de correio.

O correio é enviado entre agentes de transferência de mensagens em um formato padrão. O original formato final, RFC 822, foi revisado para o atual RFC 5322 e estendido com suporte para conteúdo multimídia e texto internacional. Este esquema é chamado MIME e será discutido mais tarde. As pessoas ainda se referem ao e-mail da Internet como RFC 822,

Apesar.

Uma ideia chave no formato da mensagem é a distinção entre o **envelope** e seu conteúdo. O envelope encapsula a mensagem. Ele contém todas as informações necessárias para o transporte da mensagem, como o endereço de destino, antes

e nível de segurança, todos distintos da própria mensagem. Temas- Os agentes de transporte sage usam o envelope para o roteamento, assim como os correios. A mensagem dentro do envelope consiste em duas partes separadas: o **cabeçalho** e o **corpo**. O cabeçalho contém informações de controle para os agentes do usuário. o corpo é inteiramente para o receptor humano. Nenhum dos agentes se preocupa muito com isso. Envelopes e mensagens são ilustrados na Figura 7-8.

Página 650

626

A CAMADA DE APLICAÇÃO

INDIVÍDUO. 7

Sr. Daniel Dumkopf

18 Willow Lane

White Plains, NY 10604

United Gizmo

180 Main St

Boston, MA 02120

1 de setembro de 2010

Sinceramente

United Gizmo

Sinceramente

United Gizmo

Assunto: Fatura 1081

Caro Sr. Dumkopf,

Nossos registros de computador

mostre que você ainda tem

não pagou a fatura acima

de \$ 0,00. Envie-nos um

verifique \$ 0,00 imediatamente.

Caro Sr. Dumkopf,

Nossos registros de computador

mostre que você ainda tem

não pagou a fatura acima

de \$ 0,00. Envie-nos um

verifique \$ 0,00 imediatamente.

Nome: Sr. Daniel Dumkopf

Rua: 18 Willow Lane

Cidade: White Plains

Estado: NY

CEP: 10604

Prioridade: Urgente

Criptografia: Nenhuma

De: United Gizmo

Endereço: 180 Main St.

Local: Boston, MA 02120

Data: 1º de setembro de 2010

Assunto: Fatura 1081

Envelope

mensagem

(uma)

(b)

Corpo

Cabeçalho

En

v

fugir

⁴⁴ e

Figura 7-8. Envelopes e mensagens. (a) Correio em papel. (b) Correio eletrônico.

Vamos examinar as peças dessa arquitetura em mais detalhes, olhando para as etapas envolvidas no envio de email de um usuário para outro. Esta jornada começa com o agente do usuário.

7.2.2 O Agente do Usuário

Um agente de usuário é um programa (às vezes chamado de **leitor de e - mail**) que aceita um variedade de comandos para escrever, receber e responder a mensagens, também quanto à manipulação de caixas de correio. Existem muitos agentes de usuário populares, incluindo Google gmail, Microsoft Outlook, Mozilla Thunderbird e Apple Mail. Eles podem variar muito em sua aparência. A maioria dos agentes de usuário tem um menu ou ícone interface gráfica orientada que requer um mouse, ou uma interface de toque em menor dispositivos móveis. Agentes de usuário mais antigos, como Elm, mh e Pine, fornecem interfaces e esperar comandos de um caracter do teclado. Funcionalmente, são iguais, pelo menos para mensagens de texto.

Os elementos típicos de uma interface de agente de usuário são mostrados na Figura 7.9. Seu

O leitor de e-mail provavelmente é muito mais chamativo, mas provavelmente tem funções equivalentes.

SEC. 7,2
CORREIO ELETRÔNICO

627

Quando um agente do usuário é iniciado, geralmente apresenta um resumo das mensagens em a caixa de correio do usuário. Frequentemente, o resumo terá uma linha para cada mensagem em alguma ordem classificada. Ele destaca os campos-chave da mensagem que são extraídos de o envelope ou cabeçalho da mensagem.

Pastas de Correio
Todos os itens
Caixa de entrada
Redes
Viagem
Lixo eletrônico
Resumo da mensagem
De
trudy
Andy
djh
Amy N. Wong
guido
Lazowska
Lazowska
...
...
...
...
...
Sujeito
Nem todos os Trudys são desagradáveis
Material sobre privacidade RFID
você viu isso?
Pedido de informação
Re: aceitação de papel
Mais sobre isso
Novo relatório lançado
Recebido
Hoje
Hoje
4 de março
3 de março
3 de março
2 de março
2 de março
Pesquisa de caixa de correio
!
Um estudante
Querido professor,
Recentemente, concluí meus estudos de graduação com
distinção em uma excelente universidade. Estarei visitando o seu
Pastas de mensagens
Pesquisa
Pós-graduação?
1 de março
mensagem

Figura 7-9. Elementos típicos da interface do agente do usuário.

Sete linhas de resumo são mostradas no exemplo da Figura 7.9. As linhas usam o Campos *De* , *Assunto* e *Recebido* , nessa ordem, para exibir quem enviou a mensagem, sobre o que se trata e quando foi recebido. Todas as informações são formatadas em um de maneira amigável, em vez de exibir o conteúdo literal dos campos da mensagem, mas é baseado nos campos da mensagem. Assim, as pessoas que deixam de incluir um *assunto* campo muitas vezes descobrem que as respostas aos seus e-mails tendem a não ser as mais altas antes ity.

Muitos outros campos ou indicações são possíveis. Os ícones ao lado da mensagem assuntos na Figura 7-9 podem indicar, por exemplo, correspondência não lida (o envelope), em material anexado (o clipe de papel) e correspondência importante, pelo menos conforme julgado pelo envio er (o ponto de exclamação).

Muitas ordens de classificação também são possíveis. O mais comum é ordenar mensagens

com base na hora em que foram recebidos, os mais recentes primeiro, com alguma indicação para saber se a mensagem é nova ou já foi lida pelo usuário. Os campos no resumo e a ordem de classificação pode ser personalizada pelo usuário de acordo com suas preferências.

Os agentes do usuário também devem ser capazes de exibir as mensagens recebidas conforme necessário para que as pessoas possam ler seus e-mails. Freqüentemente, é fornecida uma pequena prévia de uma mensagem, como em Figura 7-9, para ajudar os usuários a decidir quando ler mais. As pré-visualizações podem usar ícones pequenos ou imagens para descrever o conteúdo da mensagem. Outro processamento de apresentação

Página 652

628

A CAMADA DE APLICAÇÃO INDIVÍDUO. 7

inclui a reformatação de mensagens para caber na tela e a tradução ou conversão de conteúdo para formatos mais convenientes (por exemplo, voz digitalizada para texto reconhecido).

Após a leitura de uma mensagem, o usuário pode decidir o que fazer com ela. Isto é chamada **disposição da mensagem**. As opções incluem excluir a mensagem, enviar um responder, encaminhando a mensagem para outro usuário e mantendo a mensagem para mais tarde referência. A maioria dos agentes de usuário pode gerenciar uma caixa de correio para e-mails recebidos com

várias pastas para mensagens salvas. As pastas permitem ao usuário salvar a mensagem de acordo com o remetente, tópico ou alguma outra categoria.

O arquivamento também pode ser feito automaticamente pelo agente do usuário, antes que o usuário lê as mensagens. Um exemplo comum é que os campos e conteúdos de mensagens sábios são inspecionados e usados, juntamente com o feedback do usuário sobre mensagens, para determinar se uma mensagem é provavelmente spam. Muitos ISPs e empresas executam software que rotula e-mails como importantes ou spam para que o agente do usuário

pode arquivá-lo na caixa de correio correspondente. O ISP e a empresa têm a vantagem de uma forma de ver e-mails de muitos usuários e pode ter listas de spammers conhecidos. Se houver muitos usuários acabaram de receber uma mensagem semelhante, provavelmente é spam. Por

pré-classificar e-mails recebidos como "provavelmente legítimo" e "provavelmente spam", o usuário

agente pode economizar aos usuários uma boa quantidade de trabalho separando o que é bom do lixo.

E o spam mais popular? É gerado por coleções de comprometidos computadores chamados **botnets** e seu conteúdo depende de onde você mora. Diploma falso mas são tópicos na Ásia, e medicamentos baratos e outras ofertas de produtos duvidosos são os principais

ical nos EUA. Ainda abundam as contas bancárias nigerianas não reclamadas. Comprimidos para aumentar

várias partes do corpo são comuns em todos os lugares.

Outras regras de arquivamento podem ser elaboradas pelos usuários. Cada regra especifica uma condição

e uma ação. Por exemplo, uma regra poderia dizer que qualquer mensagem recebida do chefe vai para uma pasta para leitura imediata e qualquer mensagem de um determinado a lista de discussão vai para outra pasta para leitura posterior. Várias pastas são mostradas em Fig. 7-9. As pastas mais importantes são a Caixa de Entrada, para e-mails recebidos não arquivados em outros lugares e lixo eletrônico, para mensagens que são consideradas spam.

Bem como construções explícitas como pastas, os agentes de usuário agora fornecem recursos ricos habilidades para pesquisar a caixa de correio. Esse recurso também é mostrado na Fig. 7-9. Capa de pesquisa

As capacidades permitem que os usuários encontrem mensagens rapidamente, como a mensagem sobre "onde comprar"

Vegemite " que alguém enviou no mês passado. O e-mail percorreu um longo caminho desde os dias em que era apenas transferência de arquivos. Pró-viders agora oferecem suporte rotineiro a caixas de correio com até 1 GB de e-mail armazenado que detalha as interações de um usuário por um longo período de tempo. O sofisticado tratamento de correio de agentes de usuário com pesquisa e formas automáticas de processamento é o que o torna possível gerenciar esses grandes volumes de e-mail. Para pessoas que enviam e recebem milhares de mensagens por ano, essas ferramentas são inestimáveis. Outro recurso útil é a capacidade de responder automaticamente às mensagens em alguma maneira. Uma resposta é encaminhar o e-mail recebido para um endereço diferente, para por exemplo, um computador operado por um serviço comercial de paging que envia paging ao usuário

Página 653

SEC. 7,2
CORREIO ELETRÔNICO

629

usando rádio ou satélite e exibe a linha *Assunto*: em seu pager. Estes **auto-respontentes** devem ser executados no servidor de e-mail porque o agente do usuário pode não executar todos os tempo e pode recuperar e-mail apenas ocasionalmente. Por causa desses fatores, o usuário o agente não pode fornecer uma resposta automática verdadeira. No entanto, a interface para As respostas automáticas geralmente são apresentadas pelo agente do usuário. Um exemplo diferente de resposta automática é um **agente de férias**. Isto é um programa que examina cada mensagem recebida e envia ao remetente um insípido resposta como: " Olá. Eu estou de férias. Volto no dia 24 de agosto. Falar com então. " Essas respostas também podem especificar como lidar com assuntos urgentes no provisório, outras pessoas devem entrar em contato para problemas específicos, etc. A maioria dos agentes de férias acompanhe para quem eles enviaram respostas automáticas e evite enviar mesma pessoa uma segunda resposta. Existem armadilhas com esses agentes, no entanto. Para Por exemplo, não é aconselhável enviar uma resposta automática a uma grande lista de distribuição. Passemos agora para o cenário de um usuário enviando uma mensagem para outro usuário. Um dos recursos básicos de suporte dos agentes de usuário que ainda não discutimos é composição do correio. Envolve a criação de mensagens e respostas a mensagens e enviar essas mensagens para o resto do sistema de correio para entrega. Apesar qualquer editor de texto pode ser usado para criar o corpo da mensagem, os editores geralmente são integrado com o agente do usuário para que possa fornecer assistência com o endereçamento e os vários campos de cabeçalho anexados a cada mensagem. Por exemplo, quando responder-ao enviar uma mensagem, o sistema de e-mail pode extrair o endereço do remetente da próximo e-mail e inseri-lo automaticamente no local adequado na resposta. De outros características comuns são anexar um **bloco de assinatura** ao final de uma mensagem, corrigir ortografia e computar assinaturas digitais que mostram que a mensagem é válido. As mensagens que são enviadas para o sistema de correio têm um formato padrão que deve ser criado a partir das informações fornecidas ao agente do usuário. O mais importante parte da mensagem para transferência é o envelope, e a parte mais importante do envelope é o endereço de destino. Este endereço deve estar em um formato que o agentes de transferência de mensagens podem lidar. A forma esperada de um endereço é *user @ dns-address*. Desde que estudamos DNS no início deste capítulo, não repetiremos esse material aqui. No entanto, é vale a pena notar que existem outras formas de endereçamento. Em particular, endereços **X.400** parecem radicalmente diferentes dos endereços DNS. X.400 é um padrão ISO para sistemas de tratamento de mensagens que já foi um concorrente do SMTP. O SMTP venceu com facilidade, embora os sistemas X.400 ainda sejam usados,

principalmente fora dos endereços USX400 são compostos de *attribute = value* pares separados por barras, por exemplo,
/ C = US / ST = MASSACHUSETTS / L = CAMBRIDGE / PA = 360 MEMORIAL DR./CN=KEN SMITH /
Este endereço especifica um país, estado, localidade, endereço pessoal e nome (Ken Smith). Muitos outros atributos são possíveis, então você pode enviar e-mail para

630

A CAMADA DE APLICAÇÃO INDIVÍDUO. 7

alguém cujo endereço de e-mail exato você não conhece, desde que saiba o suficiente outros atributos (por exemplo, empresa e cargo).

Embora os nomes X.400 sejam consideravelmente menos convenientes do que os nomes DNS, o problema é discutível para agentes de usuário porque eles têm apelidos amigáveis (às vezes chamados apelidos) que permitem aos usuários inserir ou selecionar o nome de uma pessoa e obter o endereço de e-mail correto. Consequentemente, geralmente não é necessário digitar de fato essas cordas estranhas.

Um último ponto que abordaremos sobre o envio de e-mails são as listas de e-mails, que permitem envie a mesma mensagem para uma lista de pessoas com um único comando. São dois opções de como a lista de discussão é mantida. Pode ser mantido localmente, por o agente do usuário. Neste caso, o agente do usuário pode apenas enviar uma mensagem separada para cada destinatário pretendido.

Alternativamente, a lista pode ser mantida remotamente em uma transferência de mensagem agente. As mensagens serão então expandidas no sistema de transferência de mensagens, que tem o efeito de permitir que vários usuários enviem para a lista. Por exemplo, se um grupo de bird watchers tem uma lista de discussão chamada *birders* instalada no agente de transferência *meadowlark.arizona.edu*, qualquer mensagem enviada para *birders@meadowlark.arizona.edu* será encaminhado para a Universidade do Arizona e expandido em mensagens individuais a todos os membros da lista de mala direta, em qualquer parte do mundo onde estejam. Usuários deste

lista de discussão não pode dizer que é uma lista de discussão. Pode muito bem ser o pessoal caixa de correio do Prof. Gabriel O. Birders.

7.2.3 Formatos de Mensagem

Agora passamos da interface do usuário para o formato das mensagens de e-mail si mesmos. As mensagens enviadas pelo agente do usuário devem ser colocadas em um formato padrão

a ser tratada pelos agentes de transferência de mensagens. Primeiro, veremos o ASCII básico e-mail usando RFC 5322, que é a revisão mais recente da mensagem original da Internet formato sage, conforme descrito no RFC 822. Depois disso, veremos o modelo de multimídia tensões ao formato básico.

RFC 5322 - O formato de mensagem da Internet

As mensagens consistem em um envelope primitivo (descrito como parte do SMTP no RFC 5321), alguns campos de cabeçalho, uma linha em branco e, em seguida, o corpo da mensagem. Cada campo de cabeçalho (logicamente) consiste em uma única linha de texto ASCII contendo o nome do campo, dois pontos e, para a maioria dos campos, um valor. O RFC 822 original foi desassinado décadas atrás e não distinguia claramente os campos do envelope do campos de cabeçalho. Embora tenha sido revisado para RFC 5322, foi totalmente refeito não foi possível devido ao seu uso generalizado. Em uso normal, o agente do usuário constrói uma mensagem e a passa para o agente de transferência de mensagem, que então usa alguns dos campos de cabeçalho para construir o envelope real, um tanto antigo mistura antiquada de mensagem e envelope.

631

Os principais campos do cabeçalho relacionados ao transporte de mensagens estão listados na Figura 7-10.

O campo *Para*: fornece o endereço DNS do destinatário principal. Tendo vários recipientes também é permitido. O campo *Cc*: fornece os endereços de qualquer receita secundária. Em termos de entrega, não há distinção entre o primário e o secundário destinatários secundários. É uma diferença inteiramente psicológica que pode ser importante para as pessoas envolvidas, mas não é importante para o sistema de correio. O termo *Cc*: (Carbon copy) é um pouco desatualizado, já que os computadores não usam papel carbono, mas está bem estabelecido. O campo *Cco*: (cópia oculta) é como o campo *Cc*: exceto que este linha é excluída de todas as cópias enviadas aos destinatários primários e secundários. Este recurso permite que as pessoas enviem cópias para terceiros sem o principal e destinatários secundários sabendo disso.

Cabeçalho

Significado

Para:

Endereço (s) de e-mail do (s) destinatário (s) principal (is)

Cc:

Endereço (s) de e-mail do (s) destinatário (s) secundário (s)

Bcc:

Endereço (s) de e-mail para cópias ocultas

De:

Pessoa ou pessoas que criaram a mensagem

Remetente:

Endereço de email do remetente real

Recebido:

Linha adicionada por cada agente de transferência ao longo da rota

Caminho de retorno:

Pode ser usado para identificar um caminho de volta ao remetente

Figura 7-10. Campos de cabeçalho RFC 5322 relacionados ao transporte de mensagens.

Os próximos dois campos, *De*: e *Remetente*: informam quem escreveu e enviou a mensagem, respectivamente. Estes não precisam ser os mesmos. Por exemplo, um executivo de negócios pode escrever uma mensagem, mas seu assistente pode ser quem realmente a transmite.

Nesse caso, o executivo seria listado no campo *De*: e o assistente em o campo *Remetente* : . O campo *De*: é obrigatório, mas o campo *Remetente*: pode ser omitido-
ted se for igual ao campo *De* : . Esses campos são necessários no caso de a mensagem
sage não pode ser entregue e deve ser devolvida ao remetente.

Uma linha contendo *Recebido*: é adicionada por cada agente de transferência de mensagem ao longo
do

maneira. A linha contém a identidade do agente, a data e hora em que a mensagem foi refeita
recebido e outras informações que podem ser usadas para depurar o sistema de roteamento.

O campo *Return-Path*: é adicionado pelo agente de transferência de mensagem final e foi
pretendia dizer como voltar ao remetente. Em teoria, essas informações podem ser
recolhidos de todos os *Recebidos*: cabeçalhos (exceto o nome do e-mail do remetente
, mas raramente é preenchido como tal e normalmente contém apenas o anúncio do remetente
vestir.

Além dos campos da Fig. 7-10, as mensagens RFC 5322 também podem conter um
variedade de campos de cabeçalho usados pelos agentes do usuário ou destinatários humanos. A
maioria

os mais comuns estão listados na Fig. 7-11. A maioria deles é autoexplicativa, então nós
não entrarei em todos eles em muitos detalhes.

632

A CAMADA DE APLICAÇÃO

INDIVÍDUO. 7

Cabeçalho

Significado

Encontro:

A data e hora em que a mensagem foi enviada

Responder a:
Endereço de e-mail para o qual as respostas devem ser enviadas
Id da mensagem:
Número exclusivo para fazer referência a esta mensagem posteriormente
Em resposta a:
Message-Id da mensagem para a qual esta é uma resposta
Referências:
Outros IDs de mensagem relevantes
Palavras-chave:
Palavras-chave escolhidas pelo usuário
Sujeito:
Breve resumo da mensagem para o display de uma linha

Figura 7-11. Alguns campos usados no cabeçalho da mensagem RFC 5322.

O campo *Responder para*: às vezes é usado quando nem a pessoa que compõe o nem a pessoa que a enviou deseja ver a resposta. Por exemplo, um gerente de marketing pode escrever uma mensagem de e-mail contando aos clientes sobre um novo produto. A mensagem é enviada por um assistente, mas o campo *Responder para*: lista o cabeçalho do departamento de vendas, que pode tirar dúvidas e anotar pedidos. Este campo é também útil quando o remetente tem duas contas de e-mail e deseja que a resposta vá para o outro.

O *Message-Id*: é um número gerado automaticamente que é usado para vincular mensagens juntas (por exemplo, quando usadas no campo *In-Reply-To* :) e para evitar duplicação de entrega.

O documento RFC 5322 diz explicitamente que os usuários têm permissão para inventar cabeçalhos adicionais para uso privado. Por convenção, desde RFC 822, estes os cabeçalhos começam com a string *X-*. É garantido que nenhum cabeçalho futuro usará nomes que começam com *X-*, para evitar conflitos entre cabeçalhos oficiais e privados.

Às vezes, alunos de graduação inventam campos como *X-Fruit-of-the-Day*: ou *Doença-X da semana*: que são legais, embora nem sempre esclarecedoras.

Após os cabeçalhos, vem o corpo da mensagem. Os usuários podem colocar o que quiserem aqui. Algumas pessoas encerram suas mensagens com assinaturas elaboradas, incluindo citações de autoridades maiores e menores, declarações políticas e declarações de todos os tipos (por exemplo, The XYZ Corporation não é responsável por minhas opiniões; na verdade, nem mesmo pode comprehendê-los).

MIME - as extensões multifuncionais de correio da Internet

Nos primeiros dias da ARPANET, o e-mail consistia exclusivamente em mensagens de texto sábios escritos em inglês e expressos em ASCII. Para este ambiente, o primeiro formato RFC 822 fez o trabalho completamente: especificou os cabeçalhos, mas deixou o conteúdo inteiramente para os usuários. Na década de 1990, o uso mundial da Internet e demanda para enviar conteúdo mais rico por meio do sistema de correio significava que essa abordagem não era mais adequado. Os problemas incluíam envio e recebimento de mensagens

em idiomas com acentos (por exemplo, francês e alemão), alfabetos não latinos (por exemplo, Hebraico e russo) ou sem alfabetos (por exemplo, chinês e japonês), bem como enviar mensagens que não contenham texto (por exemplo, áudio, imagens ou documentos binários mentos e programas).

A solução foi o desenvolvimento do **MIME (Multipurpose Internet Mail Extensões)**. É amplamente utilizado para mensagens de e-mail enviadas pela Internet, bem como para descrever o conteúdo para outros aplicativos, como navegação na web. MIME é descrito nos RFCs 2045–2047, 4288, 4289 e 2049.

A ideia básica do MIME é continuar a usar o formato RFC 822 (o ao RFC 5322 quando o MIME foi proposto), mas para adicionar estrutura à mensagem body e define regras de codificação para a transferência de mensagens não ASCII. Não

o desvio do RFC 822 permitiu que mensagens MIME fossem enviadas usando o agentes e protocolos de transferência de correio (com base no RFC 821 na época e no RFC 5321 agora).

Tudo o que precisava ser mudado eram os programas de envio e recebimento, que os usuários poderia fazer por si próprios.

O MIME define cinco novos cabeçalhos de mensagem, conforme mostrado na Figura 7-12. O primeiro de

eles simplesmente informam ao agente do usuário que está recebendo a mensagem que está lidando com um

Mensagem MIME e qual versão do MIME ela usa. Qualquer mensagem não contém uma *versão MIME*: o cabeçalho é considerado uma mensagem de texto simples em inglês (ou pelo menos um usando apenas caracteres ASCII) e é processado como tal.

Cabeçalho

Significado

Versão MIME:

Identifica a versão MIME

Descrição do conteúdo:

String legível que informa o que está na mensagem

Content-Id:

Identificador único

Content-Transfer-Encoding:

Como o corpo é envolvido para a transmissão

Tipo de conteúdo:

Tipo e formato do conteúdo

Figura 7-12. Cabeçalhos de mensagens adicionados por MIME.

O cabeçalho *Content-Description*: é uma string ASCII que diz o que está na mensagem sóbrio. Este cabeçalho é necessário para que o destinatário saiba se vale a pena decodificá-lo ler e ler a mensagem. Se a string disser "Foto do hamster da Bárbara " e a pessoa que recebe a mensagem não é um grande fã de hamster, a mensagem provavelmente ser descartado em vez de decodificado em uma fotografia colorida de alta resolução.

O cabeçalho *Content-Id*: identifica o conteúdo. Ele usa o mesmo formato do

Message-Id padrão : cabeçalho.

O *Content-Transfer-Encoding*: diz como o corpo é embalado para transmissão missão através da rede. Um problema chave na época em que o MIME foi desenvolvido era que os protocolos de transferência de e-mail (SMTP) esperavam mensagens ASCII nas quais nenhum

a linha excede 1000 caracteres. Os caracteres ASCII usam 7 bits de cada byte de 8 bits.

Dados binários, como programas executáveis e imagens, usam todos os 8 bits de cada byte, como

634

A CAMADA DE APLICAÇÃO

INDIVÍDUO. 7

fazer conjuntos de caracteres estendidos. Não havia garantia de que esses dados seriam transferidos com segurança. Portanto, algum método de transporte de dados binários que o fazia parecer um regulamento

uma mensagem de correio ASCII lar era necessária. Extensões para SMTP desde o desenvolvimento de MIME permitem que dados binários de 8 bits sejam transferidos, embora ainda hoje os dados nem sempre passam pelo sistema de correio corretamente se não estiverem codificados. MIME fornece cinco esquemas de codificação de transferência, além de uma fuga para novos esquemas - apenas no caso. O esquema mais simples é apenas mensagens de texto ASCII. ASCII caracteres usam 7 bits e podem ser transportados diretamente pelo protocolo de e-mail, desde que nenhuma linha excede 1000 caracteres.

O próximo esquema mais simples é a mesma coisa, mas usando caracteres de 8 bits, ou seja, todos os valores de 0 a 255 são permitidos. Mensagens usando 8 bits a codificação ainda deve seguir o comprimento máximo de linha padrão.

Depois, há mensagens que usam uma codificação binária verdadeira. Estes são arbitrários arquivos binários que não apenas usam todos os 8 bits, mas também não aderem aos 1000 caracteres

limite de linha. Os programas executáveis se enquadram nesta categoria. Hoje em dia, servidores de correio

pode negociar para enviar dados em codificação binária (ou de 8 bits), voltando para ASCII se ambas as extremidades não suportam a extensão.

A codificação ASCII de dados binários é chamada de **codificação base64**. Nisso esquema, grupos de 24 bits são divididos em quatro unidades de 6 bits, com cada unidade sendo enviado como um caractere ASCII legal. A codificação é "A" para 0, "B" para 1 e assim por diante,

seguido pelas 26 letras minúsculas, os 10 dígitos e, finalmente, + e / para 62 e 63, respectivamente. As sequências == e = indicam que o último grupo continha apenas 8 ou 16 bits, respectivamente. Retornos de carro e avanços de linha são ignorados, então eles podem ser inseridos à vontade no fluxo de caracteres codificados para manter as linhas curtas o suficiente. Texto binário arbitrário pode ser enviado com segurança usando este esquema, embora ineficientemente.

Essa codificação era muito popular antes que os servidores de e-mail com capacidade binária fossem amplamente implantado. Ainda é comumente visto.

Para mensagens que são quase inteiramente ASCII, mas com alguns caracteres não ASCII acters, a codificação base64 é um tanto ineficiente. Em vez disso, uma codificação conhecida como **a codificação citada para impressão** é usada. Este é apenas ASCII de 7 bits, com todos os

níveis acima de 127 codificados como um sinal de igual seguido pelo valor do personagem como dois

dígitos hexadecimais. Caracteres de controle, alguns sinais de pontuação e símbolos matemáticos ols, bem como espaços à direita também são codificados.

Finalmente, quando há razões válidas para não usar um desses esquemas, é possível especificar uma codificação definida pelo usuário no *Content-Transfer-Encoding*: cabeçalho.

O último cabeçalho mostrado na Fig. 7-12 é realmente o mais interessante. É específico identifica a natureza do corpo da mensagem e teve um impacto muito além do e-mail. Para exemplo, o conteúdo baixado da Web é rotulado com tipos MIME para que o navegador saiba como apresentá-lo. O conteúdo é enviado por streaming de mídia e transportes em tempo real, como voz sobre IP.

Inicialmente, sete tipos de MIME foram definidos na RFC 1521. Cada tipo tem um ou mais subtipos disponíveis. O tipo e o subtipo são separados por uma barra, como em

Página 659

SEC. 7,2

CORREIO ELETRÔNICO

635

"Content-Type: video / mpeg ". Desde então, centenas de subtipos foram adicionados, junto com outro tipo. Entradas adicionais estão sendo adicionadas o tempo todo como novas tipos de conteúdo são desenvolvidos. A lista de tipos e subtipos atribuídos é principal fornecido online pela IANA em www.iana.org/assignments/media-types.

Os tipos, junto com exemplos de subtipos comumente usados, são fornecidos em Fig. 7-13. Vamos examiná-los brevemente, começando com o *texto*. O *texto / comp simple* bination é para mensagens comuns que podem ser exibidas como recebidas, sem encodificação e nenhum processamento adicional. Esta opção permite que mensagens comuns sejam transportado em MIME com apenas alguns cabeçalhos extras. O subtipo *text / html* era adicionado quando a Web se tornou popular (em RFC 2854) para permitir que as páginas da Web sejam

enviado por e-mail RFC 822. Um subtipo para eXtensible Markup Language, *text / xml*, é definido no RFC 3023. Os documentos XML proliferaram com o desenvolvimento da Web. Vamos estudar HTML e XML em Sec. 7.3.

Tipo

Subtipos de exemplo

Descrição

texto

simples, html, xml, css
Texto em vários formatos
imagem
gif, jpeg, tiff
As fotos
audio
básico, MPEG, MP4
Sons
vídeo
mpeg, mp4, quicktime
Filmes
modelo
vrml
Modelo 3D
inscrição
fluxo de octeto, pdf, javascript, zip
Dados produzidos por aplicativos
mensagem
http, rfc822
Mensagem encapsulada
multiparte

mista, alternativa, paralela, digerir Combinação de vários tipos

Figura 7-13. Tipos de conteúdo MIME e subtipos de exemplo.

O próximo tipo de MIME é *imagem*, que é usado para transmitir imagens estáticas. Muitos formatos são amplamente usados para armazenar e transmitir imagens hoje em dia, ambos com e sem compressão. Vários deles, incluindo GIF, JPEG e TIFF, são integrado em quase todos os navegadores. Muitos outros formatos e subtipos correspondentes existem também.

Os tipos de *áudio* e *vídeo* são para som e imagens em movimento, respectivamente.

Observe que o *vídeo* pode incluir apenas as informações visuais, não o som. E se um filme com som deve ser transmitido, as porções de vídeo e áudio podem ter que ser transmitido separadamente, dependendo do sistema de codificação usado. O primeiro video formato definido foi aquele criado pela modestamente chamada Moving Picture Grupo de especialistas (MPEG), mas outros foram adicionados desde então. Além de *áudio / básico*, um novo tipo de áudio, *áudio / mpeg*, foi adicionado ao RFC 3003 para permitir que as pessoas

ple para enviar arquivos de áudio MP3 por e-mail. Os tipos *video / mp4* e *audio / mp4* sinalizam vídeo

e dados de áudio armazenados no formato MPEG 4 mais recente.

O tipo de *modelo* foi adicionado após os outros tipos de conteúdo. É destinado a descrevendo dados do modelo 3D. No entanto, não foi amplamente usado até o momento.

Página 660

636

A CAMADA DE APLICAÇÃO

INDIVÍDUO. 7

O tipo de *aplicativo* é abrangente para formatos que não são cobertos por um dos os outros tipos e que requerem um aplicativo para interpretar os dados. Nós temos lis- usaram os subtipos *pdf*, *javascript* e *zip* como exemplos para documentos PDF, Java- Programas de script e arquivos Zip, respectivamente. Os agentes do usuário que recebem esta con tenda usar uma biblioteca de terceiros ou programa externo para exibir o conteúdo; o dis- o play pode ou não parecer integrado ao agente do usuário.

Ao usar tipos MIME, os agentes de usuário ganham extensibilidade para lidar com novos tipos do conteúdo do aplicativo conforme ele é desenvolvido. Este é um benefício significativo. No outro lado, muitas das novas formas de conteúdo são executadas ou interpretadas por aplicativos ções, o que apresenta alguns perigos. Obviamente, rodando um executável arbitrário programa que chegou através do sistema de e-mail de " amigos " representa um risco de segurança ard. O programa pode causar todos os tipos de danos desagradáveis às partes do computador para aos quais tem acesso, especialmente se puder ler e gravar arquivos e usar a rede.

Menos obviamente, os formatos de documento podem representar os mesmos perigos. Isto é porque formatos como PDF são linguagens de programação completas disfarçadas. Enquanto

eles são interpretados e restritos em escopo, bugs no interpretador muitas vezes permitem documentos tortuosos para escapar das restrições.

Além desses exemplos, existem muitos mais subtipos de aplicativos porque existem muitos outros aplicativos. Como um substituto para ser usado quando nenhum outro subtipo é conhecido por ser mais adequado, o subtipo de *fluxo de octeto* denota uma sequência de bytes interpretados. Ao receber tal fluxo, é provável que um agente do usuário exiba-o sugerindo ao usuário que seja copiado para um arquivo. Processo subsequente o essing fica a cargo do usuário, que presumivelmente sabe que tipo de conteúdo é. Os dois últimos tipos são úteis para compor e manipular mensagens deles-eus. O tipo de *mensagem* permite que uma mensagem seja totalmente encapsulada dentro de um de outros. Este esquema é útil para encaminhar e-mail, por exemplo. Quando um com a mensagem RFC 822 completa é encapsulada dentro de uma mensagem externa, o sub-*rfc822* tipo deve ser usado. Da mesma forma, é comum que documentos HTML sejam encapsulados sulated. E o subtipo *parcial* torna possível quebrar uma mensagem encapsulada sage em pedaços e os envie separadamente (por exemplo, se a mensagem encapsulada sage é muito longo). Os parâmetros tornam possível remontar todas as partes no destino na ordem correta.

Por fim, o tipo *multiparte* permite que uma mensagem contenha mais de uma parte, com o início e o fim de cada parte claramente delimitados. O *misto* sub-type permite que cada parte seja um tipo diferente, sem imposição de estrutura adicional. Muitos programas de e-mail permitem que o usuário forneça um ou mais anexos a um texto mensagem. Esses anexos são enviados usando o tipo *multiparte*.

Em contraste com o *misto*, o subtipo *alternativo* permite que a mesma mensagem seja incluído várias vezes, mas expresso em duas ou mais mídias diferentes. Para ex-amplamente, uma mensagem pode ser enviada em ASCII simples, em HMTL e em PDF. A propriamente

o agente de usuário projetado recebendo tal mensagem iria exibi-la de acordo com o usuário preferências. Provavelmente PDF seria a primeira escolha, se isso for possível. O segundo escolha seria HTML. Se nada disso for possível, o ASCII simples

Página 661

SEC. 7,2

CORREIO ELETRÔNICO

637

o texto seria exibido. As peças devem ser encomendadas da mais simples para a mais completa plex para ajudar os destinatários com agentes de usuário pré-MIME a entender a mensagem sage (por exemplo, até mesmo um usuário pré-MIME pode ler texto ASCII plano).

O subtipo *alternativo* também pode ser usado para vários idiomas. Neste con-texto, a Pedra de Roseta pode ser considerada uma mensagem *multiparte / alternativa inicial* sábio.

Dos outros dois subtipos de exemplo, o subtipo *paralelo* é usado quando todas as partes deve ser " visto " simultaneamente. Por exemplo, filmes geralmente têm um áudio canal e um canal de vídeo. Os filmes são mais eficazes se esses dois canais forem reproduzido em paralelo, em vez de consecutivamente. O subtipo *digest* é usado quando várias mensagens são compactadas em uma mensagem composta. Por exemplo, alguns grupos de discussão na Internet coletam mensagens de assinantes e

em seguida, envie-os para o grupo periodicamente como uma única mensagem *multipart / digest*.

Como um exemplo de como os tipos MIME podem ser usados para mensagens de e-mail, um multi-a mensagem da mídia é mostrada na Figura 7-14. Aqui, uma saudação de aniversário é transmitida em

formulários alternativos como HTML e como um arquivo de áudio. Supondo que o receptor tenha áudio

capacidade, o agente do usuário reproduzirá o arquivo de som. Neste exemplo, o som é transportado por referência como um subtipo de *mensagem / corpo externo*, então primeiro o agente do usuário

deve buscar o arquivo de som *birthday.snd* usando FTP. Se o agente do usuário não tiver áudio capacidade, as letras são exibidas na tela em silêncio pedregoso. As duas partes

são delimitados por dois hifens seguidos por uma string (gerada por software) especificada no parâmetro de *limite*.

Observe que o cabeçalho *Content-Type* ocorre em três posições dentro deste amplo. No nível superior, indica que a mensagem tem várias partes. Dentro cada parte, fornece o tipo e o subtipo dessa parte. Finalmente, dentro do corpo de a segunda parte, é necessário informar ao agente do usuário que tipo de arquivo externo é buscar. Para indicar esta ligeira diferença no uso, usamos letras minúsculas aqui, embora todos os cabeçalhos não façam distinção entre maiúsculas e minúsculas. The *Content-Transfer-En-*

a codificação é igualmente necessária para qualquer corpo externo que não seja codificado como 7 bits ASCII.

7.2.4 Transferência de Mensagem

Agora que descrevemos agentes de usuário e mensagens de e-mail, estamos prontos para veja como os agentes de transferência de mensagens retransmitem mensagens do originador para o destinatário. A transferência de correio é feita com o protocolo SMTP.

A maneira mais simples de mover mensagens é estabelecer uma conexão de transporte da máquina de origem para a máquina de destino e, em seguida, basta transferir a mensagem sóbrio. É assim que o SMTP funcionava originalmente. Com o passar dos anos, no entanto, duas diferentes usos de SMTP foram diferenciados. O primeiro uso é o **envio de email**, etapa 1 na arquitetura de e-mail da Figura 7-7. Este é o meio pelo qual o usuário agentes enviam mensagens para o sistema de correio para entrega. O segundo uso é para transferir mensagens entre agentes de transferência de mensagens (etapa 2 na Figura 7-7). Isto

Página 662

638

A CAMADA DE APLICAÇÃO

INDIVÍDUO. 7

De: alice@cs.washington.edu

Para: bob@ee.uwa.edu.au

Versão MIME: 1.0

Id da mensagem: <0704760941.AA00747@cs.washington.edu>

Tipo de conteúdo: multiparte / alternativa; limite = qwertyuiopasdfghjklzxcvbnm

Assunto: A Terra orbita o Sol número integral de vezes

Este é o preâmbulo. O agente do usuário o ignora. Tenha um bom dia.

--qwertyuiopasdfghjklzxcvbnm

Tipo de conteúdo: text / html

<p> Parabéns para você

Parabéns pra você

Feliz aniversário, querido Bob

Parabéns pra você </p>

--qwertyuiopasdfghjklzxcvbnm

Tipo de conteúdo: mensagem / corpo externo;

access-type = "anon-ftp";

site = "bicicleta.cs.washington.edu";

diretório = "pub";

nome = "aniversario.snd"

tipo de conteúdo: áudio / básico

content-transfer-encoding: base64

--qwertyuiopasdfghjklzxcvbnm--

Figura 7-14. Uma mensagem multiparte contendo HTML e alternativas de áudio.

a sequência entrega a correspondência desde o envio até a recepção da mensagem agente de transferência em um salto. A entrega final é realizada com diferentes protocolos que descreveremos na próxima seção.

Nesta seção, descreveremos os fundamentos do protocolo SMTP e seus mecanismo de tensão. Em seguida, discutiremos como ele é usado de forma diferente para sub-transferência de missão e mensagem.

SMTP (protocolo de transferência de correio simples) e extensões

Na Internet, o e-mail é entregue com o computador de envio estabelecido

Faça uma conexão TCP com a porta 25 do computador receptor. Ouvindo esta porta é um servidor de correio que fala **SMTP** (**Simple Mail Transfer Protocol**). Este ser-

ver aceita conexões de entrada, sujeito a algumas verificações de segurança e aceita mensagens para entrega. Se uma mensagem não puder ser entregue, um relatório de erro contra reter a primeira parte da mensagem não entregue é devolvida ao remetente. SMTP é um protocolo ASCII simples. Isso não é uma fraqueza, mas uma característica. O uso de texto ASCII torna os protocolos fáceis de desenvolver, testar e depurar. Eles podem ser

Página 663

SEC. 7,2
CORREIO ELETRÔNICO

639

testado enviando comandos manualmente, e os registros das mensagens são fáceis de ler. A maioria dos protocolos de Internet em nível de aplicativo agora funcionam dessa maneira (por exemplo, HTTP).

Percorreremos uma simples transferência de mensagens entre servidores de e-mail que de- transmite uma mensagem. Depois de estabelecer a conexão TCP com a porta 25, o envio máquina, operando como o cliente, espera pela máquina receptora, operando como o servidor, para falar primeiro. O servidor começa enviando uma linha de texto dando sua identidade e informando se está preparado para receber correspondência. Se não for, o cliente libera a conexão e tente novamente mais tarde.

Se o servidor estiver disposto a aceitar e-mail, o cliente anuncia a quem o e-mail está vindo e para quem vai. Se tal destinatário existir no destino ção, o servidor autoriza o cliente a enviar a mensagem. Então o cliente envia a mensagem e o servidor a confirma. Nenhuma soma de verificação é necessária ser- porque o TCP fornece um fluxo de bytes confiável. Se houver mais e-mail, agora é enviei. Quando todos os e-mails foram trocados em ambas as direções, a conexão é liberado. Um exemplo de diálogo para enviar a mensagem da Figura 7-14, incluindo o os códigos numéricos usados pelo SMTP são mostrados na Figura 7-15. As linhas enviadas pelo cli- ente (isto é, o emissor) são marcados *C*: . Aqueles enviados pelo servidor (ou seja, o receptor) são marcadas *S*: .

O primeiro comando do cliente deve ser *HELO* . Da vari- ous abreviações de quatro caracteres para *OLÁ* , este tem inúmeras vantagens sobre seu maior concorrente. Por que todos os comandos tinham que ser de quatro caracteres perdido nas brumas do tempo.

Na Fig. 7-15, a mensagem é enviada para apenas um destinatário, portanto, apenas um *RCPT* comando é usado. Esses comandos têm permissão para enviar uma única mensagem para vários receptores ple. Cada um é individualmente reconhecido ou rejeitado. Mesmo se algum os destinatários são rejeitados (porque não existem no destino), a mensagem pode ser enviado para os outros.

Finalmente, embora a sintaxe dos comandos de quatro caracteres do cliente seja especificada rigidamente, a sintaxe das respostas é menos rígida. Apenas o código numérico realmente conta. Cada implementação pode colocar qualquer string que desejar após o código.

O SMTP básico funciona bem, mas é limitado em vários aspectos. Isso não incluem autenticação. Isso significa que o comando *FROM* no exemplo poderia fornecer qualquer endereço de remetente que desejar. Isso é muito útil para enviar spam. A- outra limitação é que o SMTP transfere mensagens ASCII, não dados binários. Isto é por que a codificação de transferência de conteúdo MIME base64 era necessária. Porém, com isso codificar a transmissão de e-mail usa largura de banda de maneira ineficiente, o que é um problema para mensagens grandes. Uma terceira limitação é que o SMTP envia mensagens não criptografadas. isto não tem criptografia para fornecer uma medida de privacidade contra olhos curiosos. Para permitir que estes e muitos outros problemas relacionados ao processamento de mensagens sejam endereçado, o SMTP foi revisado para ter um mecanismo de extensão. Este mecanismo é uma parte obrigatória do padrão RFC 5321. O uso de SMTP com extensões é denominado **ESMTP (Extended SMTP)**.

640

A CAMADA DE APLICAÇÃO

INDIVÍDUO. 7

S: 220 ee.uwa.edu.au Serviço SMTP pronto

C: HELO abcd.com

S: 250 cs.washington.edu diz olá para ee.uwa.edu.au

C: CORREIO DE: <alice@cs.washington.edu>

S: 250 remetente ok

C: RCPT PARA: <bob@ee.uwa.edu.au>

S: 250 destinatário ok

C: DATA

S: 354 Enviar e-mail; termine com "." em uma linha por si só

C: De: alice@cs.washington.edu

C: Para: bob@ee.uwa.edu.au

C: Versão MIME: 1.0

C: Message-Id: <0704760941.AA00747@ee.uwa.edu.au>

C: Tipo de conteúdo: multiparte / alternativa; limite = qwertuiopasdfghjklzxcvbnm

C: Assunto: A Terra orbita o Sol número inteiro de vezes

C:

C: Este é o preâmbulo. O agente do usuário o ignora. Tenha um bom dia.

C:

C: --qwertuiopasdfghjklzxcvbnm

C: Tipo de conteúdo: text / html

C:

C: <p> Parabéns para você

C: Parabéns pra você

C: Feliz aniversário, querido <bold> Bob </bold>

C: Parabéns pra você

C:

C: --qwertuiopasdfghjklzxcvbnm

C: Tipo de conteúdo: mensagem / corpo externo;

C:

access-type = "anon-ftp";

C:

site = "bicicleta.cs.washington.edu";

C:

diretório = "pub";

C:

nome = "aniversário.snd"

C:

C: tipo de conteúdo: áudio / básico

C: codificação de transferência de conteúdo: base64

C: --qwertuiopasdfghjklzxcvbnm

C..

S: 250 mensagem aceita

C: SAIR

S: 221 ee.uwa.edu.au fechando conexão

Figura 7-15. Enviando uma mensagem de *alice@cs.washington.edu* para *bob@ee.uwa.edu.au*.

Os clientes que desejam usar uma extensão enviam uma mensagem *EHLO* em vez de *HELO* inicialmente. Se for rejeitado, o servidor é um servidor SMTP regular e o cliente deve proceder da maneira usual. Se o *EHLO* for aceito, o servidor responde com as extensões que ele suporta. O cliente pode então usar qualquer uma dessas extensões. Diversas extensões comuns são mostradas na Figura 7.16. A figura dá a palavra-chave

SEC. 7,2

CORREIO ELETRÔNICO

641

como usado no mecanismo de extensão, junto com uma descrição da nova função nacionalidade. Não entraremos em extensões com mais detalhes.

Palavra-chave

Descrição

AUTH

Autenticação de cliente

BINARYMIME

O servidor aceita mensagens binárias
CHUNKING
O servidor aceita mensagens grandes em blocos
TAMANHO
Verifique o tamanho da mensagem antes de tentar enviar
STARTTLS
Mude para o transporte seguro (TLS; consulte o Capítulo 8)
UTF8SMTP
Endereços internacionalizados

Figura 7-16. Algumas extensões SMTP.

Para ter uma ideia melhor de como o SMTP e alguns dos outros protocolos descritos neste capítulo de trabalho, experimente-os. Em todos os casos, vá primeiro a uma máquina conectada a

a Internet. Em um sistema UNIX (ou Linux), em um shell, digite telnet mail.isp.com 25

substituindo o nome DNS do servidor de correio do seu ISP por *mail.isp.com*. Em uma vitória-sistema dows XP, clique em Iniciar, Executar e digite o comando na caixa de diálogo caixa. Em uma máquina com Vista ou Windows 7, pode ser necessário primeiro instalar o telnet programa (ou equivalente) e, em seguida, inicie-o você mesmo. Este comando estabelecerá um conexão telnet (ou seja, TCP) para a porta 25 nessa máquina. A porta 25 é o SMTP porta; veja a Figura 6-34 para as portas de outros protocolos comuns. Você provavelmente irá obter uma resposta mais ou menos assim:

Tentando 192.30.200.66 ...

Conectado a mail.isp.com

O caractere de escape é '].

220 mail.isp.com Smail # 74 pronto na quinta, 25 de setembro de 2002 13:26 +0200

As três primeiras linhas são do telnet, informando o que está fazendo. A última linha é do servidor SMTP na máquina remota, anunciando sua vontade de falar para você e aceite o e-mail. Para descobrir quais comandos ele aceita, digite SOCORRO

Deste ponto em diante, uma sequência de comando como a da Fig. 7-16 é possível se o servidor está disposto a aceitar e-mails de você.

Envio de Correio

Originalmente, os agentes do usuário eram executados no mesmo computador que a mensagem de envio

agente transferido. Nesta configuração, tudo o que é necessário para enviar uma mensagem é para o usuário

agente para falar com o servidor de correio local, usando a caixa de diálogo que acabamos de descrever.

No entanto, essa configuração não é mais o caso comum.

Página 666

642

A CAMADA DE APLICAÇÃO

INDIVÍDUO. 7

Os agentes de usuário geralmente são executados em laptops, PCs domésticos e telefones celulares. Eles não são

sempre conectado à Internet. Agentes de transferência de correio executados no ISP e na empresa servidores. Eles estão sempre conectados à Internet. Essa diferença significa que um agente do usuário em Boston pode precisar entrar em contato com seu servidor de correio regular em Seattle para enviar

uma mensagem de correio porque o usuário está viajando.

Por si só, essa comunicação remota não representa nenhum problema. É exatamente o que

Os protocolos TCP / IP são projetados para oferecer suporte. No entanto, um ISP ou empresa geralmente

não quer que nenhum usuário remoto seja capaz de enviar mensagens ao seu servidor de e-mail para ser entregue em outro lugar. O ISP ou a empresa não está executando o servidor como um público serviço. Além disso, esse tipo de **retransmissão aberta** atrai spammers. Isso é porque fornece uma maneira de lavar o remetente original e, assim, tornar a mensagem mais difícil de identificar como spam.

Dadas essas considerações, o SMTP é normalmente usado para envio de e-mail com a extensão *AUTH*. Esta extensão permite que o servidor verifique as credenciais (usuário-nome e senha) do cliente para confirmar que o servidor deve fornecer serviço de correio.

Existem várias outras diferenças na forma como o SMTP é usado para envio de correio sion. Por exemplo, a porta 587 é usada em preferência à porta 25 e o servidor SMTP pode verificar e corrigir o formato das mensagens enviadas pelo agente do usuário. Para mais informações sobre o uso restrito de SMTP para envio de correio, por favor consulte RFC 4409.

Transferência de mensagens

Assim que o agente de transferência de correio de envio recebe uma mensagem do agente do usuário, ele irá entregá-lo ao agente de transferência de correio receptor usando SMTP. Para fazer isso, o remetente usa o endereço de destino. Considere a mensagem na Fig. 7-15, endereçada para *bob@ee.uwa.edu.au*. Para qual servidor de e-mail a mensagem deve ser entregue? Para determinar o servidor de correio correto a ser contatado, o DNS é consultado. No pré seção anterior, descrevemos como o DNS contém vários tipos de registros, incluindo registro do *MX*, ou *servidor de mensagens*. Neste caso, uma consulta DNS é feita para o *MX* regista do domínio *ee.uwa.edu.au*. Esta consulta retorna uma lista ordenada de nomes e endereços IP de um ou mais servidores de e-mail.

O agente de transferência de correio de envio, em seguida, faz uma conexão TCP na porta 25 para o endereço IP do servidor de correio para alcançar o agente de transferência de correio de recebimento, e

usa SMTP para retransmitir a mensagem. O agente de transferência de correio de recebimento colocará mail para o usuário *bob* na caixa de correio correta para que Bob leia mais tarde. Isto a etapa de entrega local pode envolver mover a mensagem entre os computadores, se houver uma grande infraestrutura de correio.

Com este processo de entrega, o correio viaja desde a tradução inicial até a final fer agente em um único salto. Não há servidores intermediários na transferência de mensagens etapa. É possível, no entanto, que esse processo de entrega ocorra várias vezes.

Um exemplo que já descrevemos é quando um agente de transferência de mensagens

Página 667

SEC. 7,2

CORREIO ELETRÔNICO

643

implementa uma lista de discussão. Nesse caso, é recebida uma mensagem para a lista. É então expandido como uma mensagem para cada membro da lista que é enviada para o indivíduo endereços de membros.

Como outro exemplo de retransmissão, Bob pode ter se formado no MIT e também pode ser acessado através do endereço *bob@alum.mit.edu*. Em vez de ler e-mails no várias contas, Bob pode providenciar que a correspondência enviada para este endereço seja encaminhada para

bob@ee.uwa.edu. Neste caso, o correio enviado para *bob@alum.mit.edu* passará por dois entregas. Primeiro, ele será enviado ao servidor de correio para *alum.mit.edu*. Então será enviado para o servidor de correio para *ee.uwa.edu.au*. Cada uma dessas pernas é uma entrega separada no que diz respeito aos agentes de transferência de correio.

Outra consideração hoje em dia é o spam. Nove em cada dez mensagens enviadas hoje são spam (McAfee, 2010). Poucas pessoas querem mais spam, mas é difícil evitar porque ele se disfarça como correio normal. Antes de aceitar uma mensagem, adicionais verificações podem ser feitas para reduzir as oportunidades de spam. A mensagem para Bob foi enviado de *alice@cs.washington.edu*. O agente de transferência de correio de recebimento pode procure o agente de transferência de correio de envio no DNS. Isso permite que ele verifique se o IP ad-

vestido da outra extremidade da conexão TCP corresponde ao nome DNS. Mais gen- Geralmente, o agente de recebimento pode pesquisar o domínio de envio no DNS para ver se ele

uma política de envio de correio. Essas informações são frequentemente fornecidas no *TXT* e *SPF* registros. Isso pode indicar que outras verificações podem ser feitas. Por exemplo, e-mail enviado de *cs.washington.edu* pode sempre ser enviado do host *june.cs.washington.edu*.

Se o agente de transferência de correio de envio não for *juno*, há um problema.

Se alguma dessas verificações falhar, o e-mail provavelmente está sendo falsificado com um envio falso

endereço de ing. Nesse caso, ele é descartado. No entanto, passar nessas verificações não implica que o e-mail não é spam. As verificações apenas garantem que o e-mail parece ser vindo da região da rede de onde pretende vir. A ideia é

que os spammers devem ser forçados a usar o endereço de envio correto ao enviar enviar. Isso torna o spam mais fácil de reconhecer e excluir quando for indesejado.

7.2.5 Entrega Final

Nossa mensagem está quase entregue. Chegou à caixa de correio de Bob. Todos que resta é transferir uma cópia da mensagem para o agente de usuário de Bob para exibição. Esta é a etapa 3 da arquitetura da Figura 7-7. Esta tarefa era simples no início da Internet, quando o agente de usuário e o agente de transferência de correio eram executados no mesmo ma-

chine como processos diferentes. O agente de transferência de correio simplesmente escreveu novas mensagens

ao final do arquivo de caixa de correio, e o agente do usuário simplesmente verifica o arquivo de caixa de correio

para um novo correio.

Hoje em dia, o agente do usuário em um PC, laptop ou celular provavelmente estará em uma situação diferente

máquina diferente do ISP ou servidor de correio da empresa. Os usuários desejam ser capazes de acessar

cessar seus e-mails remotamente, de onde quer que estejam. Eles querem acessar o e-mail de trabalho, de seus PCs domésticos, de seus laptops em viagens de negócios e de cybercafés durante as chamadas férias. Eles também querem trabalhar offline,

Página 668

644

A CAMADA DE APLICAÇÃO INDIVÍDUO. 7

em seguida, reconecte-se para receber e-mails e enviar e-mails de saída. Além disso, cada o usuário pode executar vários agentes de usuário, dependendo de qual computador é conveniente para

usar no momento. Vários agentes de usuário podem estar em execução ao mesmo tempo.

Nesta configuração, o trabalho do agente do usuário é apresentar uma visão do conteúdo de a caixa de correio e para permitir que a caixa de correio seja manipulada remotamente. Vários diferentes protocolos diferentes podem ser usados para esse propósito, mas o SMTP não é um deles. SMTP é um protocolo baseado em push. Ele recebe uma mensagem e se conecta a um servidor remoto para transferir a mensagem. A entrega final não pode ser alcançada desta maneira tanto porque a caixa de correio deve continuar a ser armazenada no agente de transferência de correio e ser-

fazer com que o agente do usuário não esteja conectado à Internet no momento em que O SMTP tenta retransmitir mensagens.

IMAP - O Internet Message Access Protocol

Um dos principais protocolos usados para a entrega final é o **IMAP (Internet Protocol de acesso a mensagens)**. A versão 4 do protocolo é definida no RFC 3501.

Para usar o IMAP, o servidor de e-mail executa um servidor IMAP que escuta a porta 143. O agente do usuário executa um cliente IMAP. O cliente se conecta ao servidor e começa a emitir comandos daqueles listados na Figura 7-17.

Em primeiro lugar, o cliente iniciará um transporte seguro se algum for usado (a fim de manter as mensagens e comandos confidenciais) e, em seguida, faça login ou de outra forma autenticar-se no servidor. Uma vez conectado, existem muitos comandos para listar pastas e mensagens, busque mensagens ou mesmos partes de mensagens, marque mensagens

com sinalizadores para exclusão posterior e organizar mensagens em pastas. Para evitar confusão seção, observe que usamos o termo " pasta " aqui para ser consistente com o resto do material nesta seção, em que um usuário tem uma única caixa de correio composta de várias pastas. No entanto, na especificação IMAP, o termo *caixa de correio* é usado em vez de. Um usuário, portanto, tem muitas caixas de correio IMAP, cada uma das quais é tipicamente presente

inserido para o usuário como uma pasta.

O IMAP também possui muitos outros recursos. Ele tem a capacidade de endereçar correspondência, não por

número da mensagem, mas usando atributos (por exemplo, dê-me a primeira mensagem de Alice). As pesquisas podem ser realizadas no servidor para encontrar as mensagens que satisfaçam certos critérios para que apenas essas mensagens sejam buscadas pelo cliente.

IMAP é uma melhoria em relação a um protocolo de entrega final anterior, **POP3 (Post Protocolo do Office, versão 3)**, que é especificado no RFC 1939. POP3 é um mais simples protocolo, mas suporta menos recursos e é menos seguro no uso típico. Correio é geralmente baixado para o computador do agente do usuário, em vez de permanecer no correio servidor. Isso torna a vida mais fácil no servidor, mas mais difícil para o usuário. Não é fácil ler e-mails em vários computadores e, se o computador do agente do usuário quebrar, todos os e-mails

pode ser perdido permanentemente. No entanto, você ainda encontrará o POP3 em uso.

Protocolos proprietários também podem ser usados porque o protocolo é executado entre um servidor de correio e agente de usuário que podem ser fornecidos pela mesma empresa. Microsoft O Exchange é um sistema de correio com um protocolo proprietário.

Página 669

SEC. 7,2

CORREIO ELETRÔNICO

645

Comando

Descrição

CAPACIDADE

Recursos do servidor de lista

STARTTLS

Inicie o transporte seguro (TLS; consulte o Capítulo 8)

CONECTE-SE

Faça logon no servidor

AUTENTICAR

Faça logon com outro método

SELECIONE

Selecione uma pasta

EXAMINAR

Selecionar uma pasta somente leitura

CRIO

Crie uma pasta

EXCLUIR

Apagar uma pasta

RENOMEAR

Renomear uma pasta

SE INSCREVER

Adicionar pasta ao conjunto ativo

CANCELAR SUBSCRIÇÃO

Remover pasta do conjunto ativo

LISTA

Lista as pastas disponíveis

LSUB

Lista as pastas ativas

STATUS

Obtenha o status de uma pasta

ACRESCENTAR

Adicione uma mensagem a uma pasta

VERIFICA

Obtenha um ponto de verificação de uma pasta

BUSCAR

Receba mensagens de uma pasta

PESQUISA
Encontre mensagens em uma pasta
LOJA
Alterar sinalizadores de mensagem
CÓPIA DE
Faça uma cópia de uma mensagem em uma pasta
EXPURGAR
Remover mensagens sinalizadas para exclusão
UID
Emita comandos usando identificadores exclusivos
NOOP
Fazer nada
FECHAR
Remover mensagens sinalizadas e fechar a pasta
SAIR
Saia e feche a conexão

Figura 7-17. Comandos IMAP (versão 4).

Correio eletrônico

Uma alternativa cada vez mais popular para IMAP e SMTP para fornecer e-mail serviço é usar a Web como uma interface para enviar e receber mensagens. Amplamente Os sistemas de **Webmail** usados incluem Google Gmail, Microsoft Hotmail e Yahoo! Enviar. Webmail é um exemplo de software (neste caso, um agente de usuário de correio) que é fornecido como um serviço usando a web. Nesta arquitetura, o provedor executa servidores de e-mail normalmente para aceitar mensagens para usuários com SMTP na porta 25. No entanto, o agente do usuário é diferente. Ao invés de

Página 670

646

A CAMADA DE APLICAÇÃO INDIVÍDUO. 7

sendo um programa autônomo, é uma interface de usuário fornecida por meio de páginas da web. Isso significa que os usuários podem usar qualquer navegador que quiserem para acessar seus e-mails e enviar novas mensagens.

Ainda não estudamos a Web, mas uma breve descrição de que você pode vir de volta para é o seguinte. Quando o usuário vai para a página de e-mail do provedor, um é apresentado um formulário no qual é solicitado ao usuário um nome de login e uma senha. O nome de login e a senha são enviados ao servidor, que os valida. Se o

o login for bem-sucedido, o servidor encontra a caixa de correio do usuário e cria uma lista de páginas da Web-

o conteúdo da caixa de correio em tempo real. A página da Web é então enviada para o browser para exibição.

Muitos dos itens na página que mostra a caixa de correio são clicáveis, então os sábios podem ser lidos, excluídos e assim por diante. Para tornar a interface responsiva, a Web as páginas geralmente incluem programas JavaScript. Esses programas são executados localmente em

o cliente em resposta a eventos locais (por exemplo, cliques do mouse) e também pode baixar e fazer upload de mensagens em segundo plano, para preparar a próxima mensagem para exibição ou uma nova mensagem para envio. Neste modelo, o envio de email acontece usando os protocolos normais da Web postando dados em um URL. O servidor Web cuida de injetar mensagens no sistema de entrega de correio tradicional que temos rabiscado. Para segurança, os protocolos da Web padrão também podem ser usados. Estes protocols preocupam-se em criptografar páginas da Web, não se o conteúdo de a página da Web é uma mensagem de correio.

7.3 A WEB MUNDIAL

A Web, como a World Wide Web é popularmente conhecida, é uma arquitetura estrutura para acessar conteúdo vinculado espalhada por milhões de máquinas, todas na internet. Em 10 anos, deixou de ser uma forma de coordenar o design de experimentos de física de alta energia na Suíça para a aplicação de milhões de as pessoas pensam como sendo "a Internet". Sua enorme popularidade vem do

fato de que é fácil para iniciantes usar e fornece acesso com um rico gráfico interface para uma enorme riqueza de informações sobre quase todos os subject, de aardvarks para Zulus.

A Web começou em 1989 no CERN, o Centro Europeu de Pesquisa Nuclear. A ideia inicial era ajudar grandes equipes, muitas vezes com membros em meia dúzia ou mais países e fusos horários, colabore usando uma coleção em constante mudança de relatórios, plantas, desenhos, fotos e outros documentos produzidos por experimentos em física de partículas. A proposta de uma rede de documentos vinculados veio de O físico do CERN Tim Berners-Lee. O primeiro protótipo (baseado em texto) foi operado profissional 18 meses depois. Uma demonstração pública dada na conferência Hypertext '91 cia chamou a atenção de outros pesquisadores, o que levou Marc Andreessen ao Universidade de Illinois para desenvolver o primeiro navegador gráfico. Era chamado de mosaico e lançado em fevereiro de 1993.

Página 671

SEC. 7,3
THE WORLD WIDE WEB

647

O resto, como dizem, agora é história. O mosaico era tão popular que um ano depois Andreessen saiu para formar uma empresa, a Netscape Communications Corp., cujo objetivo era desenvolver software para a Web. Pelos próximos três anos, o Netscape Navigator e O Internet Explorer da Microsoft se envolveu em uma " guerra de navegadores ", cada um tentando capturar uma fatia maior do novo mercado adicionando freneticamente mais recursos (e portanto, mais bugs) do que o outro.

Durante as décadas de 1990 e 2000, sites e páginas da Web, como o conteúdo da Web é chamadas, cresceu exponencialmente até que houvesse milhões de sites e bilhões de páginas. Um pequeno número desses sites tornou-se extremamente popular. Esses sites e o as empresas por trás deles definem amplamente a Web como as pessoas a experimentam hoje. Ex- os amplos incluem: uma livraria (Amazon, iniciada em 1994, capitalização de mercado \$ 50 bilhões), um mercado de pulgas (eBay, 1995, US \$ 30 bilhões), pesquisa (Google, 1998, US \$ 150 bilhões) e rede social (Facebook, 2004, empresa privada avaliada em mais de US \$ 15 bilhões).

O período até 2000, quando muitas empresas da Web passaram a valer centenas de milhões de dólares durante a noite, apenas para estourar praticamente no dia seguinte, quando acabou por ser hype, até tem um nome. É a chamada **era ponto com** . Novas ideias ainda estão se tornando ricos na web. Muitos deles vêm de alunos. Para exemplo, Mark Zuckerberg era um estudante de Harvard quando começou o Facebook, e Sergey Brin e Larry Page eram alunos em Stanford quando começaram o Google. Talvez você venha com a próxima grande coisa.

Talvez você venha com a próxima grande coisa.

Em 1994, CERN e MIT assinaram um acordo estabelecendo o **W3C (World Wide Web Consortium)**, uma organização dedicada a desenvolver ainda mais a Web, padronizando protocolos e encorajando a interoperabilidade entre sites. Berners-Lee se tornou o diretor. Desde então, várias centenas de universidades e empresas aderiram ao consórcio. Embora agora existam mais livros sobre a Web do que você pode imaginar, o melhor lugar para obter informações atualizadas sobre a Web está (naturalmente) na própria Web. A página inicial do consórcio está em www.w3.org . Os leitores interessados são encaminhados para links para páginas que abrangem todos

dos numerosos documentos e atividades do consórcio.

7.3.1 Visão geral da arquitetura

Do ponto de vista dos usuários, a Web consiste em uma vasta coleção mundial ção de conteúdo na forma de **páginas da Web** , freqüentemente chamadas apenas de **páginas** . Cada página pode conter links para outras páginas em qualquer lugar do mundo. Os usuários podem seguir um clicando nele, o que os leva para a página apontada. Este processo pode ser repetido indefinidamente. A ideia de ter uma página apontando para outra, agora chamado de **hipertexto** , foi inventado por um professor visionário de engenharia elétrica do MIT

engenharia, Vannevar Bush, em 1945 (Bush, 1945). Isso foi muito antes do Internet ser inventada. Na verdade, foi antes de os computadores comerciais existirem, embora várias universidades produziram protótipos rudimentares que ocuparam grandes salas e menos potência do que uma calculadora de bolso moderna.

Página 672

648

A CAMADA DE APLICAÇÃO
INDIVÍDUO. 7

Geralmente, as páginas são visualizadas com um programa chamado **navegador**. Firefox, Internet Explorer e Chrome são exemplos de navegadores populares. O navegador busca a página solicitada, interpreta o conteúdo e exibe a página, adequadamente para- emaranhada, na tela. O conteúdo em si pode ser uma mistura de texto, imagens e para- comandos de matting, na forma de um documento tradicional, ou outras formas de conteúdo, como vídeo ou programas que produzem uma interface gráfica com a qual os usuários podem interagir.

A imagem de uma página é mostrada no lado superior esquerdo da Figura 7-18. É a página para o departamento de Ciência da Computação e Engenharia da University of Washington. Esta página mostra texto e elementos gráficos (que são geralmente muito pequenos para ler). Algumas partes da página estão associadas a links para outras páginas. Um pedaço de texto, ícone, imagem e assim por diante associado a outra página é chamado de **hiperlink**. Para seguir um link, o usuário posiciona o cursor do mouse na parte vinculada do área da página (que faz com que o cursor mude de forma) e cliques. Seguindo um link é simplesmente uma maneira de dizer ao navegador para buscar outra página. Nos primeiros dias de na Web, os links foram destacados com texto sublinhado e colorido para que iria se destacar. Hoje em dia, os criadores de páginas da Web têm maneiras de controlar o aparência de regiões vinculadas, então um link pode aparecer como um ícone ou mudar sua aparência

quando o mouse passa sobre ele. Cabe aos criadores da página fazer com que o links visualmente distintos, para fornecer uma interface utilizável.

Pedido HTTP
Base de dados
página da web
Hiperlink
Rede
navegador
Documento
www.cs.washington.edu
Programa
Resposta HTTP
servidor web
youtube.com
google-analytics.com

Figura 7-18. Arquitetura da Web.

Página 673

SEC. 7,3

THE WORLD WIDE WEB

649

Os alunos do departamento podem aprender mais seguindo um link para uma página com informações especialmente para eles. Este link é acessado clicando no círculo área. O navegador então busca a nova página e a exibe, como parcialmente mostrado em a parte inferior esquerda da Fig. 7-18. Dezenas de outras páginas estão vinculadas à primeira página além deste exemplo. Todas as outras páginas podem ser compostas de conteúdo no mesmo máquina (s) como a primeira página ou em máquinas em qualquer parte do mundo. O usuário não pode contar. A busca de página é feita pelo navegador, sem qualquer ajuda do usuário. Assim, a movimentação entre as máquinas enquanto visualiza o conteúdo é perfeita. O modelo básico por trás da exibição de páginas também é mostrado na Fig. 7-18. o navegador está exibindo uma página da Web na máquina cliente. Cada página é obtida por enviar uma solicitação para um ou mais servidores, que respondem com o conteúdo do

página. O protocolo de solicitação-resposta para buscar páginas é um programa simples baseado em texto tocol que roda sobre TCP, assim como no caso do SMTP. É chamado **HTTP** (**Protocolo de transferência de hipertexto**). O conteúdo pode ser simplesmente um documento que é ler um disco ou o resultado de uma consulta ao banco de dados e execução do programa. A página é uma **página estática** se for um documento que é o mesmo sempre que é exibido. No entanto, se foi gerado sob demanda por um programa ou contém um programa, é um **página dinâmica**.

Uma página dinâmica pode se apresentar de forma diferente cada vez que é exibida. Por exemplo, a página inicial de uma loja eletrônica pode ser diferente para cada visitante. Se um cliente de livraria comprou romances de mistério no passado, ao visitar a página principal da loja, o cliente provavelmente verá novos filmes de suspense em exibição e, enquanto um cliente mais voltado para a culinária pode ser recebido com um novo cozinheiro livros. Como o site mantém registro de quem gosta de uma história para ser contada. Mas, resumidamente, a resposta envolve cookies (mesmo para visitantes com dificuldades culinárias).

Tors). Na figura, o navegador contata três servidores para buscar as duas páginas, *cs.washington.edu*, *youtube.com* e *google-analytics.com*. O conteúdo de esses diferentes servidores são integrados para exibição no navegador. A exibição envolve um intervalo de processamento que depende do tipo de conteúdo. Além de renderizar texto e gráficos, pode envolver a reprodução de um vídeo ou a execução de um script que apresenta sua própria interface de usuário como parte da página. Nesse caso, o servidor *cs.washington.edu* fornece a página principal, o servidor *youtube.com* fornece um vídeo incorporado e o servidor *google-analytics.com* não fornece nada que o usuário possa ver, mas rastreia visitantes do site. Teremos mais a dizer sobre rastreadores mais tarde.

O lado do cliente

Vamos agora examinar o lado do navegador da Web na Figura 7.18 com mais detalhes. No essencialmente, um navegador é um programa que pode exibir uma página da Web e capturar o mouse clica em itens na página exibida. Quando um item é selecionado, o navegador segue baixa o hiperlink e busca a página selecionada.

Página 674

650

A CAMADA DE APLICAÇÃO INDIVÍDUO. 7

Quando a Web foi criada pela primeira vez, ficou imediatamente claro que ter uma página aponta para outra página da Web mecanismos necessários para nomear e localizar Páginas. Em particular, três perguntas tiveram que ser respondidas antes de uma página selecionada pode ser exibido:

1. Qual é o nome da página?
2. Onde a página está localizada?
3. Como a página pode ser acessada?

Se cada página recebesse de alguma forma um nome exclusivo, não haveria ambigüidade na identificação de páginas. No entanto, o problema não seria resolvido.

Considere um paralelo entre pessoas e páginas. Nos Estados Unidos, quase todos têm um número de previdência social, que é um identificador único, já que não há duas pessoas devem ter o mesmo. No entanto, se você estiver armado apenas com um número do seguro social, não há como encontrar o endereço do proprietário, e certamente nenhuma maneira de dizer se você deve escrever para a pessoa em inglês, espanhol ou Chinês. A Web tem basicamente os mesmos problemas.

A solução escolhida identifica as páginas de uma forma que resolve todos os três problemas em uma vez. Cada página é atribuída a um **URL** (**Uniform Resource Locator**) que efetivamente serve como o nome mundial da página. URLs têm três partes: o protocolo (também conhecido como **esquema**), o nome DNS da máquina na qual a página

está localizado, e o caminho que indica exclusivamente a página específica (um arquivo para ler ou grama para rodar na máquina). No caso geral, o caminho tem um nome hierárquico que modela uma estrutura de diretório de arquivo. No entanto, a interpretação do caminho está acima para o servidor; ele pode ou não refletir a estrutura real do diretório.

Como exemplo, o URL da página mostrada na Figura 7-18 é

<http://www.cs.washington.edu/index.html>

Este URL consiste em três partes: o protocolo (*http*), o nome DNS do host (www.cs.washington.edu) e o nome do caminho (*index.html*).

Quando um usuário clica em um hiperlink, o navegador executa uma série de etapas em para buscar a página apontada. Vamos rastrear as etapas que ocorrem quando nosso ex- um link amplo é selecionado:

1. O navegador determina o URL (vendo o que foi selecionado).
2. O navegador pede ao DNS o endereço IP do servidor www.cs.washington.edu.
3. DNS responde com 128.208.3.88.
4. O navegador faz uma conexão TCP com 128.208.3.88 na porta 80, o porta bem conhecida para o protocolo HTTP.
5. Ele envia uma solicitação HTTP solicitando a página */index.html* .

Página 675

SEC. 7,3

THE WORLD WIDE WEB

651

6. O servidor www.cs.washington.edu envia a página como uma resposta HTTP sponse, por exemplo, enviando o arquivo */index.html* .

7. Se a página incluir URLs que são necessários para exibição, o navegador obtém os outros URLs usando o mesmo processo. Neste caso, o Os URLs incluem várias imagens incorporadas também obtidas de www.cs.washington.edu, um vídeo incorporado do *youtube.com* e um script de *google-analytics.com* .

8. O navegador exibe a página */index.html* conforme aparece na Fig. 7-18.

9. As conexões TCP são liberadas se não houver outras solicitações para os mesmos servidores por um curto período.

Muitos navegadores exibem a etapa que estão executando atualmente em uma linha de status na parte inferior da tela. Desta forma, quando o desempenho é ruim, o usuário pode ver se é devido ao DNS não estar respondendo, um servidor não está respondendo ou simplesmente página transmissão em uma rede lenta ou congestionada.

O design do URL é aberto no sentido de que é simples ter navegadores usam vários protocolos para obter diferentes tipos de recursos. De fato, URLs para vários outros protocolos foram definidos. Formas ligeiramente simplificadas de os mais comuns estão listados na Fig. 7-19.

Nome

Usado para

Exemplo

http

Hipertexto (HTML)

<http://www.ee.uwa.edu/~rob/>

https

Hipertexto com segurança

<https://www.bank.com/accounts/>

ftp

FTP

<ftp://ftp.cs.vu.nl/pub/minix/README>

Arquivo

Arquivo local

[arquivo: //usr/suzanne/prog.c](file:///usr/suzanne/prog.c)

mailto

Enviando email

[mailto: JohnUser@acm.org](mailto:JohnUser@acm.org)

rtsps

Streaming de mídia
rtsp://youtube.com/montypython.mpg
trago
Chamadas multimídia
sip: eve@adversary.com
sobre
Informação do navegador
sobre: plugins

Figura 7-19. Alguns esquemas comuns de URL.

Vamos examinar rapidamente a lista. O protocolo *http* é a linguagem nativa da Web, aquele falado por servidores web. **HTTP** significa **HyperText Transfer Protocol**. Vamos examiná-lo com mais detalhes posteriormente nesta seção.

O protocolo *ftp* é usado para acessar arquivos por FTP, o programa de transferência de arquivos da Internet

tocol. O FTP é anterior à Web e está em uso há mais de três décadas.

A Web facilita a obtenção de arquivos colocados em vários servidores FTP em todo o mundo, fornecendo uma interface simples e clicável em vez de uma interface interface de linha de comando. Este melhor acesso à informação é uma das razões para o crescimento espetacular da Web.

Página 676

652

A CAMADA DE APLICAÇÃO INDIVÍDUO. 7

É possível acessar um arquivo local como uma página da Web usando o protocolo de *arquivo* ou mais simplesmente, simplesmente nomeando-o. Essa abordagem não requer um servidor.

Claro, ele funciona apenas para arquivos locais, não remotos.

O protocolo *mailto* não tem realmente o sabor de buscar páginas da Web, mas é útil de qualquer maneira. Ele permite que os usuários enviem e-mail de um navegador da web. Mais sobrancelha-

sers irá responder quando um link *mailto* é seguido, iniciando o agente de e-mail do usuário para escrever uma mensagem com o campo de endereço já preenchido.

Os protocolos *rtsp* e *sip* são para estabelecer sessões de streaming de mídia e chamadas de áudio e vídeo.

Finalmente, o protocolo *sobre* é uma convenção que fornece informações sobre o navegador. Por exemplo, seguir o link *about:plugins* fará com que a maioria dos navegadores para mostrar uma página que lista os tipos MIME que eles manipulam com extensões de navegador chamados plug-ins.

Em suma, os URLs foram projetados não apenas para permitir que os usuários naveguem no Web, mas para executar protocolos mais antigos, como FTP e e-mail, bem como protocolos mais recentes

para áudio e vídeo, e para fornecer acesso conveniente a arquivos locais e navegador em formação. Esta abordagem torna todos os programas de interface de usuário especializados para aqueles outros serviços desnecessários e integra quase todo o acesso à Internet em um único programa: o navegador da web. Se não fosse pelo fato de que essa ideia era pensado por um físico britânico que trabalhava em um laboratório de pesquisa na Suíça, poderia facilmente passar por um plano idealizado pelo departamento de publicidade de alguma empresa de software - ment.

Apesar de todas essas boas propriedades, o uso crescente da Web tornou-se um fraqueza inerente ao esquema de URL. Um URL aponta para um host específico, mas às vezes é útil referenciar uma página sem dizer simultaneamente onde é. Por exemplo, para páginas com muitas referências, é desejável ter várias cópias distantes, para reduzir o tráfego da rede. Não há como dizer: " Eu deseja a página xyz , mas não me importa onde você a consegue. "

Para resolver este tipo de problema, os URLs foram generalizados em **URIs (Uniform Resource Identifiers)**. Alguns URIs informam como localizar um recurso. Esses são os URLs. Outros URIs informam o nome de um recurso, mas não onde encontrá-lo. Estes

URIs são chamados de **URN**s (**Uniform Resource Names**). As regras para escrever URIs são fornecidos no RFC 3986, enquanto os diferentes esquemas de URI em uso são rastreados por IANA. Existem muitos tipos diferentes de URIs além dos esquemas listados em Figura 7-19, mas esses esquemas dominam a Web como ela é usada hoje.

Tipos MIME

Para ser capaz de exibir a nova página (ou qualquer página), o navegador deve subsustentar seu formato. Para permitir que todos os navegadores entendam todas as páginas da web, páginas da web são escritos em uma linguagem padronizada chamada HTML. É a língua franca do Web (por enquanto). Discutiremos isso em detalhes posteriormente neste capítulo.

Página 677

SEC. 7,3
THE WORLD WIDE WEB

653

Embora um navegador seja basicamente um interpretador HTML, a maioria dos navegadores tem vários botões e recursos para facilitar a navegação na web. A maioria tem um botão para voltar à página anterior, um botão para avançar para a próxima página (só funciona após o usuário ter voltado dela), e um botão para ir direto para a página inicial preferida do usuário. A maioria dos navegadores tem um botão ou menu item para definir um favorito em uma determinada página e outro para exibir a lista de favoritos, tornando possível revisitar qualquer um deles com apenas alguns cliques.

Como nosso exemplo mostra, as páginas HTML podem conter elementos de conteúdo rico e não simplesmente texto e hipertexto. Para maior generalidade, nem todas as páginas precisam conter HTML. Uma página pode consistir em um vídeo em formato MPEG, um documento em PDF para tapete, uma fotografia em formato JPEG, uma música em formato MP3 ou qualquer uma das centenas

de outros tipos de arquivo. Como as páginas HTML padrão podem ter links para qualquer um deles, o

navegador tem um problema quando atinge uma página que não sabe interpretar.

Em vez de tornar os navegadores cada vez maiores, construindo intérpretes para uma coleção de tipos de arquivo que cresce rapidamente, a maioria dos navegadores escolheu mais

solução geral. Quando um servidor retorna uma página, ele também retorna algumas informações sobre a página. Essas informações incluem o tipo MIME da página (veja a Fig. 7-13). As páginas do tipo *text / html* são apenas exibidas diretamente, assim como as páginas em

alguns outros tipos integrados. Se o tipo MIME não for um dos integrados, o navegador consulta sua tabela de tipos MIME para determinar como exibir a página. Esta tabela associa os tipos MIME aos visualizadores.

Existem duas possibilidades: plug-ins e aplicativos auxiliares. Um **plug-in** é um módulo de código de terceiros que é instalado como uma extensão do navegador, conforme ilustrado tratado na Fig. 7-20 (a). Exemplos comuns são plug-ins para PDF, Flash e Quick hora de renderizar documentos e reproduzir áudio e vídeo. Porque os plug-ins são executados dentro o navegador, eles têm acesso à página atual e podem modificar sua aparência.

Processo
Ajudante
inscrição
Navegador
Plug-in do navegador
Processo
Processo
(b)
(uma)

Figura 7-20. (a) Um plug-in de navegador. (b) Um aplicativo auxiliar.

Cada navegador possui um conjunto de procedimentos que todos os plug-ins devem implementar para que o navegador pode chamar os plug-ins. Por exemplo, normalmente há um procedimento que

654

A CAMADA DE APLICAÇÃO
INDIVÍDUO. 7

chamadas de código base do navegador para fornecer ao plug-in os dados a serem exibidos. Este conjunto de

procedimentos é a interface do plug-in e é específico do navegador.

Além disso, o navegador disponibiliza um conjunto de seus próprios procedimentos para o plug-in, para fornecer serviços a plug-ins. Procedimentos típicos na interface do navegador face são para alocar e liberar memória, exibindo uma mensagem no navegador do linha de status e consultar o navegador sobre os parâmetros.

Antes que um plug-in possa ser usado, ele deve ser instalado. A instalação usual procedência é para o usuário ir ao site do plug-in e baixar uma instalação

Arquivo. Executar o arquivo de instalação descompacta o plug-in e torna o apropriado chamadas para registrar o tipo MIME do plug-in com o navegador e associar o plug-in com ele. Os navegadores geralmente vêm pré-carregados com plug-ins populares.

A outra maneira de estender um navegador é usar um **aplicativo auxiliar**. Isto é um programa completo, funcionando como um processo separado. É ilustrado na Fig. 7-20 (b). Uma vez que o auxiliar é um programa separado, a interface está à distância de o navegador. Normalmente aceita apenas o nome de um arquivo de trabalho onde o conteúdo arquivo foi armazenado, abre o arquivo e exibe o conteúdo. Normalmente, ajudantes são grandes programas que existem independentemente do navegador, por exemplo, Microsoft Word ou PowerPoint.

Muitos aplicativos auxiliares usam o *aplicativo do tipo MIME*. Como consequência, um número considerável de subtipos foi definido para eles usarem, por exemplo *application / vnd.ms-powerpoint* para arquivos PowerPoint. *vnd* denota vendor-specific formatos específicos. Desta forma, um URL pode apontar diretamente para um arquivo PowerPoint, e

quando o usuário clica nele, o PowerPoint é iniciado automaticamente e entrega o conteúdo a ser exibido. Os aplicativos auxiliares não se restringem ao uso do *aplicativo* *cation* MIME type .. Adobe Photoshop usa *image / x-photoshop*, por exemplo.

Consequentemente, os navegadores podem ser configurados para lidar com uma virtualmente ilimitada

número de tipos de documentos sem alterações em si mesmos. Servidores da web modernos são frequentemente configurados com centenas de combinações de tipo / subtipo e novos são frequentemente adicionados sempre que um novo programa é instalado.

Uma fonte de conflitos é que vários plug-ins e aplicativos auxiliares são disponível para alguns subtipos, como *video / mpeg*. O que acontece é que o último um para registrar sobrescreve a associação existente com o tipo MIME, capturando o tipo por si só. Como consequência, a instalação de um novo programa pode alterar o forma como um navegador lida com os tipos existentes.

Os navegadores também podem abrir arquivos locais, sem rede à vista, em vez de buscar carregá-los de servidores Web remotos. No entanto, o navegador precisa de alguma forma de determinar o tipo MIME do arquivo. O método padrão é para o sistema operacional tentar de associar uma extensão de arquivo a um tipo MIME. Em uma configuração típica, abrir *foo.pdf* irá abri-lo no navegador usando um plug-in de *aplicativo / pdf* e abrir *bar.doc* o abrirá no Word como o auxiliar do *aplicativo / msword*.

Aqui, também, podem surgir conflitos, uma vez que muitos programas estão dispostos - não, faça tão ansioso - para lidar com, digamos, *mpg*. Durante a instalação, programas destinados a usuários sofisticados geralmente exibem caixas de seleção para os tipos e extensões MIME

SEC. 7,3
THE WORLD WIDE WEB

655

eles estão preparados para lidar para permitir que o usuário selecione os apropriados e portanto, não sobrescrever associações existentes por acidente. Programas voltados para a

mercado sumer assume que o usuário não tem ideia do que é um tipo MIME e simplesmente pegue tudo o que puderem sem levar em conta o que gramas fizeram.

A capacidade de estender o navegador com um grande número de novos tipos é conveniente, mas também pode causar problemas. Quando um navegador em um PC com Windows busca um

arquivo com a extensão *exe*, ele percebe que este arquivo é um programa executável e portanto, não tem ajudante. A ação óbvia é executar o programa. No entanto, este pode ser uma enorme falha de segurança. Tudo que um site malicioso precisa fazer é

duce uma página da Web com fotos de, digamos, estrelas de cinema ou heróis do esporte, todos os quais

estão ligados a um vírus. Um único clique em uma imagem causa um desconhecido e programa executável potencialmente hostil a ser obtido e executado na máquina do usuário.

Para evitar visitantes indesejados como este, o Firefox e outros navegadores vêm configurados ter cuidado ao executar programas desconhecidos automaticamente, mas nem todos os usuários entender quais escolhas são seguras em vez de convenientes.

O lado do servidor

Tanto para o lado do cliente. Agora, vamos dar uma olhada no lado do servidor. Como nós vimos acima, quando o usuário digita um URL ou clica em uma linha de hipertexto, o navegador analisa o URL e interpreta a parte entre *http://* e a próxima barra como um nome DNS para pesquisar. Armado com o endereço IP do servidor, o navegador estabelece uma conexão TCP com a porta 80 nesse servidor. Em seguida, ele envia uma mensagem mand contendo o resto da URL, que é o caminho para a página nesse servidor.

O servidor então retorna a página para o navegador exibir.

Para uma primeira aproximação, um servidor Web simples é semelhante ao servidor de Fig. 6-6. Esse servidor recebe o nome de um arquivo para pesquisar e retornar pela rede trabalhos. Em ambos os casos, as etapas que o servidor executa em seu loop principal são:

1. Aceite uma conexão TCP de um cliente (um navegador).
2. Obtenha o caminho para a página, que é o nome do arquivo solicitado.
3. Obtenha o arquivo (do disco).
4. Envie o conteúdo do arquivo para o cliente.
5. Libere a conexão TCP.

Os servidores da Web modernos têm mais recursos, mas, em essência, é isso que um servidor da Web

faz para o caso simples de conteúdo contido em um arquivo. Para con dinâmico tenda, a terceira etapa pode ser substituída pela execução de um programa (determinado do caminho) que retorna o conteúdo.

No entanto, os servidores da Web são implementados com um design diferente para servir a muitos solicitações por segundo. Um problema com o design simples é que o acesso aos arquivos é

Página 680

656

A CAMADA DE APLICAÇÃO
INDIVÍDUO. 7

muitas vezes, o gargalo. As leituras de disco são muito lentas em comparação com a execução do programa,

e os mesmos arquivos podem ser lidos repetidamente do disco usando chamadas do sistema operacional.

Outro problema é que apenas uma solicitação é processada por vez. O arquivo pode ser grande, e outras solicitações serão bloqueadas durante a transferência.

Uma melhoria óbvia (usada por todos os servidores Web) é manter um cache em memória dos *n* arquivos lidos mais recentemente ou um certo número de gigabytes de con- barraca. Antes de ir para o disco para obter um arquivo, o servidor verifica o cache. Se o arquivo for ali, pode ser servido diretamente da memória, eliminando assim o acesso ao disco.

Embora o cache eficaz exija uma grande quantidade de memória principal e alguns tempo extra de processamento para verificar o cache e gerenciar seu conteúdo, a economia em o tempo quase sempre compensa a sobrecarga e as despesas.

Para resolver o problema de atender a uma única solicitação por vez, uma estratégia é tornar o servidor **multithread**. Em um design, o servidor consiste em um front-end módulo que aceita todas as solicitações de entrada e módulos de processamento k , conforme mostrado em

Fig. 7-21. Os $k + 1$ threads pertencem todos ao mesmo processo, então o processamento todos os módulos têm acesso ao cache dentro do espaço de endereço do processo. Quando um a solicitação chega, o front end a aceita e cria um breve registro descrevendo-a.

Em seguida, ele entrega o registro a um dos módulos de processamento.

Em processamento
módulo
(fio)
Cache
A parte dianteira
Disco
Solicitação
Resposta
Cliente
Servidor

Figura 7-21. Um servidor Web multithread com um front end e módulos de processamento.

O módulo de processamento primeiro verifica o cache para ver se o arquivo necessário está lá. Nesse caso, ele atualiza o registro para incluir um ponteiro para o arquivo no registro. Se não é lá, o módulo de processamento inicia uma operação de disco para lê-lo no cache (possivelmente descartar de forma flexível alguns outros arquivos em cache para liberar espaço para ele). Quando o arquivo

vem do disco, é colocado no cache e também enviado de volta ao cliente.

A vantagem deste esquema é que enquanto um ou mais módulos de processamento estão bloqueados esperando a conclusão de uma operação de disco ou rede (e, portanto, supondo nenhum tempo de CPU), outros módulos podem estar trabalhando ativamente em outras solicitações.

Com k módulos de processamento, a taxa de transferência pode ser k vezes maior do que com um servidor de thread único. Claro, quando o disco ou rede é o limite

Página 681

SEC. 7,3

THE WORLD WIDE WEB

657

fator, é necessário ter vários discos ou uma rede mais rápida para obter qualquer imagem real comprovação sobre o modelo single-threaded.

Os servidores da Web modernos fazem mais do que apenas aceitar nomes de caminhos e retornar arquivos. No

Na verdade, o processamento real de cada solicitação pode ser bastante complicado. Por esta razão filho, em muitos servidores, cada módulo de processamento executa uma série de etapas. A frente final passa cada solicitação de entrada para o primeiro módulo disponível, que então carrega usando algum subconjunto das etapas a seguir, dependendo de quais são necessário para esse pedido específico. Essas etapas ocorrem após a conexão TCP e qualquer mecanismo de transporte seguro (como SSL / TLS, que será descrito no cap. 8) foram estabelecidas.

1. Resolva o nome da página da Web solicitada.
2. Execute o controle de acesso na página da web.
3. Verifique o cache.
4. Obtenha a página solicitada do disco ou execute um programa para criá-la.
5. Determine o restante da resposta (por exemplo, o tipo MIME a ser enviado).
6. Retorne a resposta ao cliente.
7. Faça uma entrada no log do servidor.

A etapa 1 é necessária porque a solicitação de entrada pode não conter o nome real do arquivo ou programa como uma string literal. Ele pode conter atalhos integrados que precisam ser traduzido. Como um exemplo simples, o URL <http://www.cs.vu.nl/> tem um vazio nome do arquivo. Deve ser expandido para algum nome de arquivo padrão que geralmente é *index.html*. Outra regra comum é mapear *~ user* / no diretório da Web do usuário .

Essas regras podem ser usadas juntas. Assim, a página inicial de um dos autores

(AST) pode ser contatado em

<http://www.cs.vu.nl/~ast/>

mesmo que o nome do arquivo real seja *index.html* em um determinado diretório padrão.

Além disso, os navegadores modernos podem especificar informações de configuração, como o software do navegador e o idioma padrão do usuário (por exemplo, italiano ou inglês). Isto torna possível para o servidor selecionar uma página da Web com pequenas imagens para um dispositivo móvel e no idioma preferido, se disponível. Em geral, nome e expansão não é tão trivial quanto pode parecer à primeira vista, devido a uma variedade de convenções e informações sobre como mapear caminhos para o diretório de arquivos e programas.

A etapa 2 verifica se alguma restrição de acesso associada à página está conhecida. Nem todas as páginas estão disponíveis para o público em geral. Determinar se um cliente pode buscar uma página pode depender da identidade do cliente (por exemplo, conforme fornecido por

nomes de usuário e senhas) ou a localização do cliente no DNS ou espaço IP.

Por exemplo, uma página pode ser restrita a usuários dentro de uma empresa. Como isso é

Página 682

658

A CAMADA DE APLICAÇÃO

INDIVÍDUO. 7

realizado depende do design do servidor. Para o popular servidor Apache, por exemplo, a convenção é colocar um arquivo chamado *.htaccess* que lista as restrições no diretório onde a página restrita está localizada.

As etapas 3 e 4 envolvem obter a página. Se pode ser retirado do cache depende das regras de processamento. Por exemplo, páginas que são criadas por programas nem sempre podem ser armazenados em cache porque eles podem produzir um resultado diferente

cada vez que eles são executados. Mesmo os arquivos devem ser ocasionalmente verificados para ver se

seus conteúdos foram alterados para que os conteúdos antigos possam ser removidos do cache. Se a página requer a execução de um programa, também há o problema de configuração dos parâmetros do programa ou entrada. Esses dados vêm do caminho ou de outras partes do pedido.

A etapa 5 é sobre como determinar outras partes da resposta que acompanham o conteúdo da página. O tipo MIME é um exemplo. Pode vir do arquivo extensão, as primeiras palavras do arquivo ou saída do programa, um arquivo de configuração, e possivelmente outras fontes.

A etapa 6 é retornar a página pela rede. Para aumentar o desempenho, uma única conexão TCP pode ser usada por um cliente e servidor para várias páginas. Essa reutilização significa que alguma lógica é necessária para mapear uma solicitação para um

conexão e para retornar cada resposta de modo que seja associada com o resultado correto da busca.

A etapa 7 faz uma entrada no log do sistema para fins administrativos, juntamente com quaisquer outras estatísticas importantes. Esses toros podem ser posteriormente minerados para obter valor

informações capazes sobre o comportamento do usuário, por exemplo, a ordem em que as pessoas visitaram as páginas.

Biscoitos

Navegar na Web como descrevemos até agora envolve uma série de buscas de páginas pendentes. Não existe o conceito de uma sessão de login. O navegador envia uma solicitação a um servidor e recebe de volta um arquivo. Então, o servidor esquece que já vi aquele cliente em particular.

Este modelo é perfeitamente adequado para recuperar documentos publicamente disponíveis, e funcionou bem quando a Web foi criada. No entanto, não é adequado para retornar páginas diferentes para usuários diferentes, dependendo do que eles já tenham feito com o servidor. Este comportamento é necessário para muitas interações contínuas com

Web sites. Por exemplo, alguns sites (por exemplo, jornais) exigem que os clientes registrem ister (e possivelmente pagar dinheiro) para usá-los. Isso levanta a questão de como ser- vers podem distinguir entre solicitações de usuários que se cadastraram anteriormente e todos os outros. Um segundo exemplo vem do e-commerce. Se um usuário vagar em uma loja de eletrônicos, jogando itens em seu carrinho de compras virtual de vez em quando ao tempo, como o servidor controla o conteúdo do carrinho? Um terceiro ex- amplo são portais da Web personalizados, como o Yahoo!. Os usuários podem configurar um

Página 683

SEC. 7,3
THE WORLD WIDE WEB

659

página inicial detalhada com apenas as informações que eles desejam (por exemplo, seus estoques e seus times esportivos favoritos), mas como o servidor pode exibir a página correta se não sabe quem é o usuário?

À primeira vista, pode-se pensar que os servidores poderiam rastrear os usuários observando seus endereços IP. No entanto, essa ideia não funciona. Muitos usuários compartilham computadores, especialmente em casa, e o endereço IP apenas identifica o computador, não o usuário. Pior ainda, muitas empresas usam NAT, de modo que os pacotes de saída suportam o mesmo endereço IP para todos os usuários. Ou seja, todos os computadores por trás do NAT a caixa tem a mesma aparência para o servidor. E muitos ISPs atribuem endereços IP aos clientes com DHCP. Os endereços IP mudam com o tempo, então para um servidor você pode se parecer com seu vizinho. Por todos esses motivos, o servidor não pode usar um anúncio de IP vestidos para rastrear usuários.

Esse problema é resolvido com um mecanismo frequentemente criticado chamado **cookies**. O nome deriva da antiga gíria do programador em que um programa chama um procedimento dure e receba algo que possa precisar apresentar mais tarde para conseguir algum trabalho feito. Nesse sentido, um descritor de arquivo UNIX ou um identificador de objeto do Windows pode ser

considerado um cookie. Os cookies foram implementados pela primeira vez no Netscape browser em 1994 e agora são especificados no RFC 2109.

Quando um cliente solicita uma página da Web, o servidor pode fornecer informações adicionais informação na forma de um cookie juntamente com a página solicitada. O cookie é um string nomeada bem pequena (de no máximo 4 KB) que o servidor pode associar a um navegador. Essa associação não é a mesma coisa que um usuário, mas é muito mais próxima e mais útil do que um endereço IP. Os navegadores armazenam os cookies oferecidos para um intervalo, geralmente em um diretório de cookies no disco do cliente para que os cookies persistam a invocações entre navegadores, a menos que o usuário tenha desativado os cookies. Cookies são apenas

strings, não programas executáveis. Em princípio, um cookie pode conter um vírus, mas uma vez que os cookies são tratados como dados, não existe uma forma oficial para o vírus realmente

correr e causar danos. No entanto, é sempre possível para algum hacker explorar um bug do navegador para causar ativação.

Um cookie pode conter até cinco campos, conforme mostrado na Figura 7.22. O *Domínio* diz de onde o cookie veio. Os navegadores devem verificar se os servidores não estão mentindo sobre seu domínio. Cada domínio não deve armazenar mais de 20 cookies por cliente. O *caminho* é um caminho na estrutura de diretório do servidor que identifica quais partes da árvore de arquivos do servidor podem usar o cookie. Muitas vezes é /, que significa a árvore inteira.

O campo *Conteúdo* assume o formato *nome = valor*. Tanto o *nome* quanto o *valor* podem ser tudo o que o servidor quiser. Este campo é onde o conteúdo do cookie é armazenado.

O campo *Expires* especifica quando o cookie expira. Se este campo estiver ausente, o navegador descarta o cookie ao sair. Esse cookie é chamado de **não persistente**.

biscoito consistente. Se a hora e a data são fornecidas, o cookie é dito ser um **persistente**. O cookie consistente é mantido até que expire. Os tempos de expiração são fornecidos em Horário de Greenwich. Para remover um cookie do disco rígido de um cliente, um servidor

apenas o envia novamente, mas com um tempo de expiração no passado.

660

A CAMADA DE APLICAÇÃO

INDIVÍDUO. 7

Dominio

Caminho

Conteúdo

Expira

Seguro

toms-casino.com

/

CustomerID = 297793521

15-10-10 17:00

sim

jills-store.com

/

Carrinho = 1-00501; 1-07031; 2-13721

11-1-11 14:22

Não

aportal.com

/

Prefs = Stk: CSCO + ORCL; Spt: Jets

31-12-20 23:59

Não

sneaky.com

/

UserID = 4627239101

31-12-19 23:59

Não

Figura 7-22. Alguns exemplos de cookies.

Finalmente, o campo *Seguro* pode ser definido para indicar que o navegador só pode voltar a transformar o cookie em um servidor usando um transporte seguro, ou seja, SSL / TLS (que nós descreveremos no cap. 8). Este recurso é usado para comércio eletrônico, bancos e outros aplicativos seguros.

Já vimos como os cookies são adquiridos, mas como são usados? Apenas seja-antes que um navegador envie uma solicitação de uma página para algum site, ele verifica o seu cookie di-

reitoria para ver se algum cookie foi colocado pelo domínio que o pedido está indo para. Se assim for, todos os cookies colocados por esse domínio, e apenas esse domínio, são incluído na mensagem de solicitação. Quando o servidor os obtém, ele pode interpretá-los da maneira que ele quiser.

Vamos examinar alguns usos possíveis para cookies. Na Figura 7-22, o primeiro cookie foi definido por *toms-casino.com* e é usado para identificar o cliente. Quando o cliente retorna na próxima semana para jogar fora mais algum dinheiro, o navegador envia o cookie para que o servidor saiba quem é. Armado com o ID do cliente, o servidor pode procurar o registro do cliente em um banco de dados e usar essas informações para construir um página da Web apropriada para exibição. Dependendo do jogo conhecido do cliente hábitos, esta página pode consistir em uma mão de pôquer, uma lista das corridas de cavalos de hoje ou

uma máquina caça-níqueis.

O segundo cookie veio de *jills-store.com*. O cenário aqui é que o cliente está vagando pela loja, procurando coisas boas para comprar. Quando ela encontra uma pechincha e clica nela, o servidor a adiciona ao carrinho de compras (mantido no servidor) e também cria um cookie contendo o código do produto do item e envia o cookie de volta para o cliente. Conforme o cliente continua a vagar pelo loja clicando em novas páginas, o cookie é devolvido ao servidor a cada novo solicitação de página. Conforme mais compras se acumulam, o servidor as adiciona ao cookie. Finalmente, quando o cliente clica em PROCEED TO CHECKOUT , o cookie, agora contendo a lista completa de compras, é enviada junto com a solicitação. Desta forma, o servidor sabe exatamente o que o cliente deseja comprar.

O terceiro cookie é para um portal da web. Quando o cliente clica em um link para o portal, o navegador envia o cookie. Isso diz ao portal para construir uma página contendo os preços das ações da Cisco e da Oracle, e do futebol do New York Jets resultados. Como um cookie pode ter até 4 KB, há bastante espaço para mais detalhes preferências relacionadas a manchetes de jornais, clima local, ofertas especiais, etc.

Página 685

SEC. 7.3
THE WORLD WIDE WEB

661

Um uso mais controverso de cookies é rastrear o comportamento online dos usuários. Isso permite que os operadores de sites entendam como os usuários navegam em seus sites e os anunciantes criam perfis dos anúncios ou sites que um determinado usuário visualizou. O polêmica é que os usuários normalmente não sabem que suas atividades estão sendo rastreadas, mesmo com perfis detalhados e em sites aparentemente não relacionados. No entanto, o rastreamento da Web é um grande negócio. DoubleClick, que fornece e rastreia anúncios, está classificado entre os 100 sites mais movimentados do mundo pelo monitoramento da Web empresa Alexa. O Google Analytics, que rastreia o uso do site para operadores, é usado por mais da metade dos 100.000 sites mais ocupados da web.

É fácil para um servidor rastrear a atividade do usuário com cookies. Suponha um servidor quer acompanhar quantos visitantes únicos teve e quantas páginas cada visitante olhou antes de sair do site. Quando a primeira solicitação chega, não haverá nenhum cookie de acompanhamento, então o servidor envia de volta um cookie com *Contador de retenção = 1*. As visualizações de página subsequentes nesse site enviarão o cookie de volta ao servidor. Cada vez que o contador é incrementado e enviado de volta ao cliente. Ao controlar os contadores, o servidor pode ver quantas pessoas dão

depois de ver a primeira página, quantos olham para duas páginas e assim por diante.

Rastrear o comportamento de navegação dos usuários em sites é apenas um pouco mais complicado. Funciona assim. Uma agência de publicidade, digamos, Sneaky Ads, atrai os principais sites da Web e coloca anúncios dos produtos de seus clientes em suas páginas, para

que paga aos proprietários do site uma taxa. Em vez de dar aos sites o anúncio como um arquivo GIF para colocar em cada página, dá a eles um URL para adicionar a cada página. Cada URL é

hands out contém um número único no caminho, como

<http://www.sneaky.com/382674902342.gif>

Quando um usuário visita pela primeira vez uma página, P , que contém tal anúncio, o navegador busca

o arquivo HTML. Em seguida, o navegador inspeciona o arquivo HTML e vê o link para o arquivo de imagem em www.sneaky.com, para enviar uma solicitação de imagem lá. Um GIF arquivo contendo um anúncio é retornado, junto com um cookie contendo um ID de usuário exclusivo,

4627239101 na Fig. 7-22. Sneaky registra o fato de que o usuário com este ID página visitada P . Isso é fácil de fazer, pois o caminho solicitado ([382674902342.gif](http://www.sneaky.com/382674902342.gif)) é referenciada apenas na página P . Claro, o anúncio real pode aparecer em milhares de páginas, mas cada vez com um nome diferente. Sneaky provavelmente coleta uma fração de um centavo do fabricante do produto cada vez que ele envia o anúncio.

Mais tarde, quando o usuário visita outra página da Web contendo qualquer um dos anúncios da Sneaky,

o navegador primeiro busca o arquivo HTML do servidor. Em seguida, ele vê o link para, digamos, <http://www.sneaky.com/193654919923.gif> na página e solicite esse arquivo.

Como já tem um cookie do domínio *sneaky.com*, o navegador inclui cookie do Sneaky contendo o ID do usuário. Sneaky agora conhece uma segunda página do usuário visitou.

No devido tempo, Sneaky pode construir um perfil detalhado da navegação do usuário hábitos, mesmo que o usuário nunca tenha clicado em nenhum dos anúncios. Claro que ainda não tem o nome do usuário (embora tenha seu endereço IP, que

662

A CAMADA DE APLICAÇÃO
INDIVÍDUO. 7

pode ser o suficiente para deduzir o nome de outros bancos de dados). No entanto, se o usuário sempre fornece seu nome a qualquer site que coopera com Sneaky, um perfil completo junto com um nome estará disponível para venda para quem quiser comprá-lo. A venda dessas informações pode ser lucrativa o suficiente para que a Sneaky coloque mais anúncios em sites e, assim, coletar mais informações.

E se o Sneaky quiser ser superneaky, o anúncio não precisa ser um banner clássico de Anúncios. Um "anúncio" consistindo em um único pixel na cor de fundo (e portanto invisível) tem exatamente o mesmo efeito que um anúncio de banner: requer que o navegador vá buscar a imagem GIF de 1 × 1 pixel e enviar para ela todos os cookies originados no domínio Principal.

Os cookies tornaram-se um ponto focal para o debate sobre a privacidade online porque de rastreamento de comportamento como o acima. A parte mais insidiosa de todo o negócio é que muitos usuários desconhecem completamente esta coleta de informações e podem até acham que estão seguros porque não clicam em nenhum dos anúncios. Por esta razão, os cookies que rastreiam os usuários em sites são considerados por muitos como **spyware**.

Dê uma olhada nos cookies que já estão armazenados em seu navegador. Mais sobrancelhadoras exibirá essas informações junto com as preferências de privacidade atuais. Você pode se surpreender ao encontrar nomes, endereços de e-mail ou senhas, bem como opacos identificadores. Felizmente, você não encontrará números de cartão de crédito, mas o potencial para o abuso é claro.

Para manter uma aparência de privacidade, alguns usuários configuraram seus navegadores para rejeitar todos os cookies. No entanto, isso pode causar problemas porque muitos sites não funcionarão corretamente sem cookies. Como alternativa, a maioria dos navegadores permite que os usuários

bloquear **cookies de terceiros**. Um cookie de terceiros é aquele de um site diferente da página principal que está sendo buscada, por exemplo, o cookie *sneaky.com* que é usado ao interagir com a página *P* em um site completamente diferente. Bloqueando esses cookies ajudam a evitar o rastreamento em sites. As extensões do navegador podem também ser instaladas para fornecer controle refinado sobre como os cookies são usados (ou, em vez disso, não usado). Conforme o debate continua, muitas empresas estão desenvolvendo privacidade políticas que limitam como eles irão compartilhar informações para evitar abusos. Claro, as políticas são simplesmente como as empresas dizem que tratarão as informações. Para exemplo: "Podemos usar as informações coletadas de você na condução de nossos negócios" - que pode estar vendendo as informações.

7.3.2 Páginas da Web estáticas

A base da Web é a transferência de páginas da Web do servidor para o cliente. De forma mais simples, as páginas da Web são estáticas. Ou seja, elas são apenas arquivos em algum servidor que se apresentam da mesma maneira cada vez que são buscados e vistos. Só porque são estáticos não significa que as páginas sejam inertes no navegador, no entanto. Uma página que contém um vídeo pode ser uma página da Web estática. Como mencionado anteriormente, a língua franca da Web, em que a maioria das páginas são escritas, é HTML. As páginas iniciais dos professores são geralmente páginas HTML estáticas.

SEC. 7,3

THE WORLD WIDE WEB

663

As páginas iniciais das empresas são geralmente páginas dinâmicas reunidas por um site da companhia de design. Nesta seção, daremos uma breve olhada nas páginas HTML estáticas como base para material posterior. Os leitores já familiarizados com HTML podem pular adiante para a próxima seção, onde descrevemos o conteúdo dinâmico e os serviços da Web.

HTML - a linguagem de marcação de hipertexto

O HTML (HyperText Markup Language) foi introduzido na web. isto permite aos usuários produzir páginas da Web que incluem texto, gráficos, vídeo, ponteiros para outras páginas da Web e muito mais. HTML é uma linguagem de marcação, ou linguagem para descrevendo como os documentos devem ser formatados. O termo " marcação " vem de os velhos tempos, quando os editores de texto marcavam documentos para informar à impressora - naquela época, um ser humano - quais fontes usar e assim por diante. Linguagens de marcação portanto, contém comandos explícitos para formatação. Por exemplo, em HTML, **** significa iniciar o modo negrito e **** significa sair do modo negrito. LaTeX e TeX são outros exemplos de linguagens de marcação que são bem conhecidas pela maioria autores acadêmicos.

A principal vantagem de uma linguagem de marcação sobre outra sem marcação explícita é que separa o conteúdo de como deve ser apresentado. Escrever um navegador é então direto: o navegador simplesmente precisa entender os comandos de marcação e aplicá-los ao conteúdo. Incorporando todos os comandos de marcação em cada Arquivo HTML e padronizá-los torna possível para qualquer navegador da Web ler e reformate qualquer página da web. Isso é crucial porque uma página pode ter sido produzido em uma janela de 1600×1200 com cores de 24 bits em um computador de última geração, mas pode

deve ser exibido em uma janela de 640×320 em um telefone celular.

Embora seja certamente possível escrever documentos como este com qualquer texto simples editor, e muitas pessoas fazem, também é possível usar processadores de texto ou Editores de HTML que fazem a maior parte do trabalho (mas, correspondentemente, dão ao usuário menos

controle direto sobre os detalhes do resultado final).

Uma página da Web simples escrita em HTML e sua apresentação em um navegador são dado na Fig. 7-23. Uma página da Web consiste em um cabeçalho e um corpo, cada um delimitado por tags `<html>` e `</html>` (comandos de formatação), embora a maioria dos navegadores não reclamar se essas tags estiverem faltando. Como pode ser visto na Fig. 7-23 (a), a cabeça é entre colchetes pelas tags `<head>` e `</head>` e o corpo está entre colchetes pelo tags `<body>` e `</body>`. As strings dentro das tags são chamadas de **diretivas**. A maioria, mas não todas, as tags HTML têm este formato. Ou seja, eles usam `<algo>` para marcar o começo de algo e `</something>` para marcar seu fim.

As tags podem estar em minúsculas ou maiúsculas. Assim, `<head>` e `<HEAD>` significa a mesma coisa, mas minúsculas é melhor para compatibilidade. Layout real do O documento HTML é irrelevante. Os analisadores HTML ignoram os espaços extras e o carro retorna, uma vez que eles precisam reformatar o texto para ajustá-lo à área de exibição atual. Consequentemente, o espaço em branco pode ser adicionado à vontade para tornar os documentos HTML mais

legível, algo de que a maioria deles precisa desesperadamente. Como outra consequência, linhas em branco não podem ser usadas para separar parágrafos, pois são simplesmente ignoradas. A tag explícita é necessária.

Algumas tags têm parâmetros (nomeados), chamados **atributos**. Por exemplo, o A tag `` da Fig. 7-23 é usada para incluir uma imagem alinhada com o texto. Tem dois atributos, `src` e `alt`. O primeiro atributo fornece o URL da imagem. O padrão HTML não especifica quais formatos de imagem são permitidos. Na prática, todos os navegadores suportam arquivos GIF e JPEG. Os navegadores são gratuitos para oferecer suporte a outros tapetes, mas esta extensão é uma espada de dois gumes. Se um usuário está acostumado a uma sobrancelha ser que suporta, digamos, arquivos TIFF, ele pode incluí-los em suas páginas da Web e posteriormente

ficará surpreso quando outros navegadores simplesmente ignorarem toda a sua arte maravilhosa. O segundo atributo fornece um texto alternativo para usar se a imagem não puder ser exibida reproduziu. Para cada tag, o padrão HTML fornece uma lista do que o pa- os parâmetros, se houver, são e o que significam. Como cada parâmetro é nomeado, o a ordem em que os parâmetros são fornecidos não é significativa.

Tecnicamente, os documentos HTML são escritos no ISO 8859-1 Latin-1 character definido, mas para usuários cujos teclados suportam apenas ASCII, as sequências de escape são presente para os caracteres especiais, como e` . A lista de caracteres especiais é fornecida no padrão. Todos eles começam com um "e" comercial e terminam com um ponto e vírgula.

Por exemplo, produz um espaço, è produz e` e é ; prôduces é. Uma vez que <,> e & têm significados especiais, eles podem ser expressos apenas com suas sequências de escape, < ;, > ; e & ;, respectivamente.

O item principal no cabeçalho é o título, delimitado por <title> e </title> . Certo tipos de metainformação também podem estar presentes, embora nenhum esteja presente em nosso ex

amplio. O título em si não é exibido na página. Alguns navegadores usam para rotular a janela da página.

Vários cabeçalhos são usados na Figura 7.23. Cada título é gerado por um <h n > tag, onde n é um dígito no intervalo de 1 a 6. Assim, <h1> é o cabeçalho mais importante-ing; <h6> é o menos importante. Depende do navegador processar esses aplicativos apropriadamente na tela. Normalmente, os títulos de numeração mais baixa serão dis-tocado em uma fonte maior e mais pesada. O navegador também pode optar por usar diferentes cores para cada nível de título. Normalmente, os títulos <h1> são grandes e em negrito com pelo menos uma linha em branco acima e abaixo. Em contraste, <h2> cabeçalhos estão em um fonte menor com menos espaço acima e abaixo.

As marcas e <i> são usadas para entrar no modo negrito e itálico, respectivamente.

A tag <hr> força uma quebra e desenha uma linha horizontal na tela.

A tag <p> inicia um parágrafo. O navegador pode exibir isso inserindo um linha em branco e algum recuo, por exemplo. Curiosamente, a tag </p> que existe para marcar o final de um parágrafo é muitas vezes omitido por preguiçoso HTML pro grammers.

HTML fornece vários mecanismos para fazer listas, incluindo listas aninhadas.

Listas não ordenadas, como as da Figura 7-23, são iniciadas com , com usado para marcar o início dos itens. Também existe uma tag para iniciar uma lista ordenada. o

Página 689

SEC. 7,3

THE WORLD WIDE WEB

665

```
<html>
<head> <title> AMALGAMATED WIDGET, INC. </title> </head>
<body> <h1> Bem-vindo à página inicial do AWI </h1>
<img src = "http://www.widget.com/images/logo.gif" ALT = "AWI Logo"> <br>
Estamos muito felizes por você ter escolhido visitar o <b> Amalgamated Widget's </b>
pagina inicial. Esperamos que <i> você </i> encontre todas as informações de que precisa aqui.
<p> Abaixo, temos links para informações sobre nossos diversos produtos finos.
Você pode fazer o pedido eletronicamente (por WWW), por telefone ou por e-mail. </p>
<hr>
<h2> Informações do produto </h2>
<ul>
<li> <a href="http://widget.com/products/big"> Big widgets </a> </li>
<li> <a href="http://widget.com/products/little"> Pequenos widgets </a> </li>
</ul>
<h2> Informações de contato </h2>
<ul>
<li> Por telefone: 1-800-WIDGETS </li>
<li> Por e-mail: info@amalgamated-widget.com </li>
</ul>
</body>
</html>
(uma)
```

Bem-vindo à página inicial da AWI

Estamos muito felizes por você ter optado por visitar a página inicial do **Amalgamated Widget**. Nós esperamos que você encontrará todas as informações de que precisa aqui. Abaixo, temos links para informações sobre nossos muitos produtos finos. Você pode fazer o pedido eletronicamente (por WWW), por telefone ou por e-mail.

informação do produto

- Widgets grandes

- Pequenos widgets

Informações de Contato

- Por telefone: 1-800-WIDGETS

- Por e-mail: info@amalgamated-widget.com

(b)

Figura 7-23. (a) O HTML para uma página da Web de amostra. (b) A página formatada.

Página 690

666

A CAMADA DE APLICAÇÃO

INDIVÍDUO. 7

itens individuais em listas não ordenadas geralmente aparecem com marcadores () na frente deles. Os itens nas listas ordenadas são numerados pelo navegador.

Finalmente, chegamos aos hiperlinks. Exemplos destes são vistos na Fig. 7-23 usando as tags `<a>` (âncora) e ``. A tag `<a>` tem vários parâmetros, os mais importantes do qual é `href` o URL vinculado. O texto entre `<a>` e `` é reproduzido. Se estiver selecionado, o hiperlink é seguido para uma nova página. Também é permitido

ted para ligar outros elementos. Por exemplo, uma imagem pode ser fornecida entre `<a>` e `` tags usando ``. Neste caso, a imagem é exibida e clicando sobre ela ativa o hiperlink.

Existem muitas outras tags e atributos HTML que não vimos neste exemplo simples. Por exemplo, a tag `<a>` pode receber um *nome* de parâmetro para plantar um hyperlink, permitindo que um hyperlink aponte para o meio de uma página. Isso é útil, por exemplo, para páginas da Web que começam com um índice clicável. Por clicando em um item no índice, o usuário salta para o correspondente seção da mesma página. Um exemplo de uma tag diferente é `
`. Isso força o navegador para quebrar e iniciar uma nova linha.

Provavelmente, a melhor maneira de entender as tags é vê-las em ação. Façam isso, você pode escolher uma página da Web e olhar o HTML em seu navegador para ver como a página foi montada. A maioria dos navegadores tem um item de menu VIEW SOURCE (ou alguma coisa similar). Selecionar este item exibe o código-fonte HTML da página atual, em vez de sua saída formatada.

Nós esboçamos as marcas que existiam desde o início da Web. HTML continua evoluindo. A Fig. 7-24 mostra alguns dos recursos que foram adicionados com versões sucessivas de HTML. HTML 1.0 refere-se à versão de HTML usada com a introdução da Web. As versões HTML 2.0, 3.0 e 4.0 apareceram em sucessão rápida no espaço de apenas alguns anos, enquanto a Web explodia. Depois de HTML 4.0, um período de quase dez anos se passou antes do caminho para a padronização da próxima versão principal, HTML 5.0, ficou claro. Porque é uma grande atualização que consolida as maneiras como os navegadores lidam com conteúdo rico, o esforço do HTML 5.0 está em andamento e não se espera que produza um padrão antes de 2012, no mínimo.

Apesar dos padrões, os principais navegadores já suportam a função HTML 5.0 nacionalidade.

A progressão através das versões HTML envolve a adição de novos recursos que

as pessoas queriam, mas tinham que lidar de maneiras não padronizadas (por exemplo, plug-ins) até que tornou-se padrão. Por exemplo, HTML 1.0 e HTML 2.0 não tinham tabelas. Eles foram adicionados em HTML 3.0. Uma tabela HTML consiste em uma ou mais linhas, cada uma consistindo em uma ou mais células de tabela que podem conter uma ampla gama de materiais (por exemplo, texto, imagens, outras tabelas). Antes do HTML 3.0, os autores precisavam de uma tabela teve que recorrer a métodos ad hoc, como incluir uma imagem mostrando a mesa. No HTML 4.0, mais novos recursos foram adicionados. Isso incluía acessibilidade recursos para usuários com deficiência, incorporação de objetos (uma generalização do tag para que outros objetos também possam ser incorporados nas páginas), suporte para linguagem de script indicadores (para permitir conteúdo dinâmico) e muito mais.

Página 691

SEC. 7,3
THE WORLD WIDE WEB

667

Item

HTML 1.0

HTML 2.0

HTML 3.0

HTML 4.0

HTML 5.0

Hiperlinks

x

x

x

x

x

Imagens

x

x

x

x

x

Listas

x

x

x

x

x

Mapas e imagens ativos

x

x

x

x

Formulários

x

x

x

x

Equações

x

x

x

Barras de Ferramentas

x

x

x

Mesas

x

x

x

Recursos de acessibilidade

x
x
Incorporação de objetos
x
x
Folhas de estilo
x
x
Scripting
x
x
Video e audio
x
Gráficos vetoriais embutidos
x
Representação XML
x
Tópicos de fundo
x
Armazenamento do navegador
x
Tela de desenho
x

Figura 7-24. Algumas diferenças entre as versões HTML.

HTML 5.0 inclui muitos recursos para lidar com a mídia rica que agora é rotineira minimamente usado na Web. Vídeo e áudio podem ser incluídos nas páginas e reproduzidos por o navegador sem exigir que o usuário instale plug-ins. Desenhos podem ser construídos no navegador como gráficos vetoriais, em vez de usar formatos de imagem bitmap (como JPEG e GIF). Também há mais suporte para a execução de scripts em navegadores, como threads de computação em segundo plano e acesso ao armazenamento. Todos esses recursos ajuda a oferecer suporte a páginas da Web que são mais como aplicativos tradicionais com um usuário interface do que documentos. Esta é a direção que a Web está tomando.

Entrada e formulários

Há um recurso importante que ainda não discutimos: entrada. O HTML 1.0 era basicamente unilateral. Os usuários podem buscar páginas de informações providers, mas era difícil enviar informações de outra maneira. É rápido tornou-se evidente que havia uma necessidade de tráfego de mão dupla para permitir pedidos para produtos a serem colocados por meio de páginas da Web, cartões de registro a serem preenchidos online, termos de pesquisa a serem inseridos e muito, muito mais.

Página 692

668

A CAMADA DE APLICAÇÃO INDIVÍDUO. 7

O envio de dados do usuário para o servidor (por meio do navegador) requer dois tipos de suporte. Primeiro, requer que o HTTP seja capaz de transportar dados nessa direção. Nós descreva como isso é feito em uma seção posterior; ele usa o método *POST*. O segundo requisito é ser capaz de apresentar elementos de interface do usuário que reúnem e embalam envelhecer a entrada. **Os formulários** foram incluídos com essa funcionalidade no HTML 2.0. Os formulários contêm caixas ou botões que permitem aos usuários preencher informações ou fazer escolhas e enviar as informações de volta para o proprietário da página. Os formulários são escritos dez, exatamente como outras partes do HTML, conforme mostrado no exemplo da Figura 7.25. Observe que formulários ainda são conteúdo estático. Eles exibem o mesmo comportamento, independentemente de quem é usando-os. O conteúdo dinâmico, que abordaremos mais tarde, oferece mais sofisticadas citadas de coletar dados enviando um programa cujo comportamento pode depender de o ambiente do navegador.

Como todos os formulários, este está entre as tags <form> e </form> . o

atributos desta tag dizem o que fazer com os dados de entrada, neste caso usando o método *POST* para enviar os dados ao URL especificado. Texto não incluído em um tag é apenas exibida. Todas as tags usuais (por exemplo, ****) são permitidas em um formulário para permitir que o

autor da página controle a aparência do formulário na tela.

Três tipos de caixas de entrada são usados neste formulário, cada um dos quais usa o tag **<input>**. Possui uma variedade de parâmetros para determinar o tamanho, natureza e uso da caixa exibida. Os formulários mais comuns são campos em branco para aceitar o texto do usuário, as caixas que podem ser marcadas e os botões de *envio* que causam o dados a serem devolvidos ao servidor.

O primeiro tipo de caixa de entrada é uma caixa de *texto* que segue o texto "Nome". o caixa tem 46 caracteres de largura e espera que o usuário digite uma string, que é armazenado na variável *cliente*.

A próxima linha do formulário pede o endereço do usuário, 40 caracteres Largo. Em seguida, vem uma linha perguntando sobre a cidade, estado e país. Uma vez que nenhuma tag **<p>**

são usados entre esses campos, o navegador exibe todos eles em uma linha (em vez como parágrafos separados) se eles caberem. No que diz respeito ao navegador, o um parágrafo contém apenas seis itens: três strings alternando com três caixas.

A próxima linha pede o número do cartão de crédito e a data de validade. Transmitindo números de cartão de crédito na Internet só devem ser feitos quando houver segurança adequada medidas foram tomadas. Discutiremos alguns deles no Cap. 8

Após a data de expiração, encontramos um novo recurso: botões de opção.

Eles são usados quando uma escolha deve ser feita entre duas ou mais alternativas. o modelo intelectual aqui é um rádio de carro com meia dúzia de botões para escolher stações. Clicar em um botão desativa todos os outros no mesmo grupo. o apresentação visual depende do navegador. O tamanho do widget também usa dois botões de opção.

Os dois grupos são diferenciados por seus parâmetros de *nome*, não por escopo estático usando algo como **<radiobutton> ... </radiobutton>**.

Os parâmetros de *valor* são usados para indicar qual botão de opção foi pressionado. Por exemplo, dependendo de quais opções de cartão de crédito o usuário escolheu, o a variável *cc* será definida como a string "mastercard" ou a string "visacard".

Página 693

SEC. 7,3

THE WORLD WIDE WEB

669

```
<html>
<head> <title> FORMULÁRIO DE PEDIDO DO CLIENTE AWI </title> </head>
<body>
<h1> Formulário de pedido do widget </h1>
<form ACTION = "http://widget.com/cgi-bin/order.cgi" method = POST>
<p> Nome <input name = "customer" size = 46> </p>
<p> Endereço <input name = "address" size = 40> </p>
<p> Cidade <input name = "city" size = 20> Estado <input name = "state" size = 4>
País <input name = "country" size = 10> </p>
<p> Cartão de crédito # <input name = "cardno" size = 10>
Expira em <input name = "expires" size = 4>
M / C <input name = "cc" type = radio value = "mastercard">
VISA <input name = "cc" type = radio value = "visacard"> </p>
<p> Tamanho do widget Grande <input name = "product" type = radio value = "expensive">
Pequeno <input name = "product" type = radio value = "cheap">
Envie por correio expresso <input name = "express" type = checkbox> </p>
<p> <input type = submit value = "Submit order"> </p>
Obrigado por solicitar um widget AWI, o melhor widget que o dinheiro pode comprar!
</form>
</body>
</html>
(uma)
```

Formulário de pedido de widget

```

Nome
Endereço
Cidade
Cartão de crédito #
Tamanho do widget Grande
Obrigado por solicitar um widget AWI, o melhor widget que o dinheiro pode comprar!
Pequeno
Envie por correio expresso
Expira
M / C
Visto
Estado
País
Enviar pedido
(b)

```

Figura 7-25. (a) O HTML para um formulário de pedido. (b) A página formatada.

Após os dois conjuntos de botões de opção, chegamos à opção de envio, representada por uma caixa de *seleção* de tipo `. Pode ser ativado ou desativado. Ao contrário dos botões de rádio, onde exatamente um do conjunto deve ser escolhido, cada caixa de seleção de tipo pode estar ligado ou desligado, independentemente de todos os outros.`

Página 694

670

A CAMADA DE APLICAÇÃO INDIVÍDUO. 7

Finalmente, chegamos ao botão de *envio*. A string de *valor* é o rótulo no botão e é exibido. Quando o usuário clica no botão *enviar*, o navegador empacota as informações coletadas em uma única linha longa e as envia de volta para o servidor ao URL fornecido como parte da tag `<form>`. Uma codificação simples é usada. O `&` é usado para separar campos e `+` é usado para representar o espaço. Para nosso exemplo forma, a linha pode ser semelhante ao conteúdo da Figura 7.26.

```

cliente = John + Doe & address = 100 + Main + St. & city = White + Plains &
state = NY & country = USA & cardno = 1234567890 & expira = 6/14 & cc = mastercard &
produto = barato & expresso = em

```

Figura 7-26. Uma possível resposta do navegador ao servidor com informações informações preenchidas pelo usuário.

A string é enviada de volta ao servidor como uma linha. (Está dividido em três linhas aqui porque a página não é larga o suficiente.) Cabe ao servidor dar sentido a isso string, provavelmente passando as informações para um programa que irá processá-la. Discutiremos como isso pode ser feito na próxima seção.

Existem também outros tipos de entrada que não são mostrados neste exemplo simples. Dois outros tipos são *senha* e *textarea*. Uma caixa de *senha* é o mesmo que um *texto* caixa (o tipo padrão que não precisa ser nomeado), exceto que os caracteres não são exibidos à medida que são digitados. A *textarea* caixa também é o mesmo que um *texto* caixa, com exceção

que pode conter várias linhas.

Para listas longas nas quais uma escolha deve ser feita, o `<select>` e `</select>` tags são fornecidas para incluir uma lista de alternativas. Esta lista é frequentemente processada como um

menu suspenso. A semântica é a dos botões de opção, a menos que os *múltiplos* parameter é fornecido, caso em que a semântica é aquela das caixas de seleção.

Finalmente, existem maneiras de indicar valores padrão ou iniciais que o usuário pode mudar. Por exemplo, se uma caixa de *texto* recebe um campo de *valor*, o conteúdo é distribuído reproduzido no formulário para o usuário editar ou apagar.

CSS - Cascading Style Sheets

O objetivo original do HTML era especificar a *estrutura* do documento, não sua *aparência*. Por exemplo,

```

<h1> Fotos de Deborah </h1>

```

instrui o navegador a enfatizar o título, mas não diz nada sobre o tipo de letra, o tamanho do ponto ou a cor. Isso é deixado para o navegador, que conhece o

propriedades da tela (por exemplo, quantos pixels ele possui). No entanto, muitos designers de páginas queriam controle absoluto sobre como suas páginas eram exibidas, tão novas tags foram adicionadas ao HTML para controlar a aparência, como
 Fotos de Deborah

Página 695

SEC. 7,3
THE WORLD WIDE WEB

671

Além disso, formas foram adicionadas para controlar o posicionamento na tela com precisão. O problema

O que há de mais comum nessa abordagem é que ela é entediante e produz HTML inchado que não é portátil. Embora uma página possa renderizar perfeitamente no navegador em que foi desenvolvida, pode ser uma bagunça completa em outro navegador ou outra versão do mesmo navegador ou em uma resolução de tela diferente.

Uma alternativa melhor é o uso de folhas de estilo. Folhas de estilo em editores de texto permitem autores associarem o texto a um estilo lógico em vez de um estilo físico, por exemplo, "parágrafo inicial" em vez de "texto em itálico." A aparência de cada estilo é definido separadamente. Desta forma, se o autor decidir alterar o parágrafo inicial gráficos de itálico de 14 pontos em azul a negrito de 18 pontos em rosa choque, tudo isso requires está mudando uma definição para converter o documento inteiro.

CSS (Cascading Style Sheets) introduziu folhas de estilo na Web com HTML 4.0, embora o uso generalizado e o suporte ao navegador não tenham decolado até 2000. CSS define uma linguagem simples para descrever regras que controlam a aparência do conteúdo etiquetado. Vamos ver um exemplo. Suponha que AWI deseja páginas da Web elegantes com texto marinho na fonte Arial em um fundo esbranquiçado, e títulos de nível que são 100% e 50% maiores do que o texto para cada nível, respectivamente. A definição CSS na Figura 7.27 fornece essas regras.

```
corpo {cor de fundo: linho; cor: marinho; família da fonte: Arial;}  
h1 {tamanho da fonte: 200%;}  
h2 {tamanho da fonte: 150%;}
```

Figura 7-27. Exemplo de CSS.

Como pode ser visto, as definições de estilo podem ser compactas. Cada linha seleciona um elemento ao qual se aplica e dá os valores das propriedades. As propriedades de um elemento se aplicam como padrão a todos os outros elementos HTML que ele contém. Então, o style for body define o estilo dos parágrafos do texto no corpo. Também há con abreviações venientes para nomes de cores (por exemplo, vermelho). Quaisquer parâmetros de estilo que não sejam definidos são preenchidos com padrões pelo navegador. Este comportamento torna a folha de estilo definições opcionais; alguma apresentação razoável ocorrerá sem eles.

As folhas de estilo podem ser colocadas em um arquivo HTML (por exemplo, usando a tag <style>), mas é mais comum colocá-los em um arquivo separado e fazer referência a eles. Por exemplo, a tag <head> da página AWI pode ser modificada para se referir a uma folha de estilo no arquivo *awistyle.css* conforme mostrado na Figura 7-28. O exemplo também mostra o tipo MIME de arquivos CSS para ser *texto / css*.

```
<head>  
<title> AMALGAMATED WIDGET, INC. </title>  
<link rel = "stylesheet" type = "text / css" href = "awistyle.css" />  
</head>
```

Figura 7-28. Incluindo uma folha de estilo CSS.

Página 696

672

A CAMADA DE APLICAÇÃO
INDIVÍDUO. 7

Essa estratégia tem duas vantagens. Primeiro, permite que um conjunto de estilos seja aplicado a

muitas páginas em um site. Esta organização dá uma aparência consistente para páginas, mesmo que tenham sido desenvolvidas por autores diferentes em momentos diferentes, e todos

permite que a aparência de todo o site seja alterada editando um arquivo CSS e não o HTML. Este método pode ser comparado a um arquivo #include em um programa C: a definição de uma macro muda em todos os arquivos de programa que incluem o cabeçalho. A segunda vantagem é que os arquivos HTML baixados são mantido pequeno. Isso ocorre porque o navegador pode baixar uma cópia do arquivo CSS para todas as páginas que fazem referência a ele. Não é necessário baixar uma nova cópia do definido juntamente com cada página da web.

7.3.3 Páginas da Web e Aplicativos da Web Dinâmicos

O modelo de página estática que usamos até agora trata as páginas como documentos multimídia que estão convenientemente ligados entre si. Foi um modelo adequado no início dias da Web, à medida que grandes quantidades de informações eram colocadas online. Hoje em dia, muito do entusiasmo em torno da Web é usado para aplicativos e serviços.

Os exemplos incluem a compra de produtos em sites de comércio eletrônico, pesquisa em catálogos de bibliotecas

logs, explorando mapas, lendo e enviando e-mail e colaborando em documentos.

Esses novos usos são como software de aplicativo tradicional (por exemplo, leitores de e-mail e processadores de texto). A diferença é que esses aplicativos são executados dentro do navegador, com dados de usuário armazenados em servidores em centros de dados da Internet. Eles usam protocolos da Web

para acessar informações pela Internet e o navegador para exibir uma interface de usuário.

A vantagem desta abordagem é que os usuários não precisam instalar aplicativos separados programas de cação e dados do usuário podem ser acessados de diferentes computadores e apoiado pelo operador de serviço. Está provando ser tão bem sucedido que rivaliza software de aplicação tradicional. Claro, o fato de que esses aplicativos são oferecido gratuitamente por grandes provedores ajuda. Este modelo é a forma predominante de **nuvem**

computação, em que a computação sai de computadores desktop individuais e entra clusters compartilhados de servidores na Internet.

Para atuar como aplicativos, as páginas da Web não podem mais ser estáticas. Conteúdo dinâmico é necessário. Por exemplo, uma página do catálogo da biblioteca deve refletir quais livros são atualmente disponíveis e quais livros foram retirados e, portanto, não estão disponíveis.

Da mesma forma, uma página útil do mercado de ações permitiria ao usuário interagir com a página para ver os preços das ações em diferentes períodos de tempo e calcular os lucros e perdas. Como esses exemplos sugerem, o conteúdo dinâmico pode ser gerado por

gramas rodando no servidor ou no navegador (ou em ambos os lugares).

Nesta seção, examinaremos cada um desses dois casos separadamente. O general situação é mostrada na Figura 7.29. Por exemplo, considere um serviço de mapa que permite o usuário insere um endereço e apresenta um mapa correspondente do local.

Dado um pedido de localização, o servidor da Web deve usar um programa para criar uma página que mostra o mapa para a localização de um banco de dados de ruas e outros informações gráficas. Esta ação é mostrada nas etapas 1 a 3. A solicitação (etapa

7
2
6
Rede
página
Programa
4
DB

Figura 7-29. Páginas dinâmicas.

No entanto, há mais conteúdo dinâmico. A página que é retornada pode contém programas que são executados no navegador. Em nosso exemplo de mapa, o programa permitiria ao usuário encontrar rotas e explorar áreas próximas em diferentes níveis de detalhe. Ele atualizaria a página, aumentando ou diminuindo o zoom conforme orientado pelo usuário (etapa 4). Para

lidar com algumas interações, o programa pode precisar de mais dados do servidor. No neste caso, o programa irá enviar uma solicitação ao servidor (passo 5) que irá recuperar mais informações do banco de dados (etapa 6) e retornar uma resposta (etapa 7). o o programa continuará atualizando a página (etapa 4). Os pedidos e respostas acontecer em segundo plano; o usuário pode nem estar ciente deles porque o URL e o título da página normalmente não mudam. Incluindo programas do lado do cliente, a página pode apresentar uma interface mais responsiva do que com programas do lado do servidor sozinho.

Geração de página da web dinâmica no servidor

Vejamos o caso da geração de conteúdo do lado do servidor com mais detalhes. Um sim-
A situação em que o processamento do lado do servidor é necessário é o uso de formulários.
Considere o usuário preenchendo o formulário de pedido AWI da Figura 7-25 (b) e clicando no Botão *Enviar pedido*. Quando o usuário clica, uma solicitação é enviada ao servidor no URL especificado com o formulário (um *POST* para <http://widget.com/cgi-bin/order.cgi> em neste caso) junto com o conteúdo do formulário conforme preenchido pelo usuário. Estes dados deve ser fornecido a um programa ou script para processar. Assim, o URL identifica o pro-
grama para correr; os dados são fornecidos ao programa como entrada. Neste caso, proc-
essing envolveria inserir o pedido no sistema interno da AWI, atualizando registros do mer, e cobrando o cartão de crédito. A página retornada por esta solicitação irá dependem do que acontece durante o processamento. Não é corrigido como uma página estática. E se

Se o pedido for bem-sucedido, a página retornada pode fornecer a data de envio esperada. Se isso for malsucedido, a página retornada pode dizer que os widgets solicitados estão fora do estoque ou o cartão de crédito não era válido por algum motivo.

Página 698

674

A CAMADA DE APLICAÇÃO INDIVÍDUO. 7

Exatamente como o servidor executa um programa em vez de recuperar um arquivo depende de o design do servidor web. Não é especificado pelos próprios protocolos da Web.

Isso ocorre porque a interface pode ser proprietária e o navegador não precisa conheça os detalhes. No que diz respeito ao navegador, ele está simplesmente fazendo uma solicitação e buscar uma página.

No entanto, APIs padrão foram desenvolvidas para servidores Web para invocar programas. A existência dessas interfaces torna mais fácil para os desenvolvedores ex-
cuidar de diferentes servidores com aplicativos da Web. Veremos brevemente duas APIs para dar-lhe uma ideia do que eles implicam.

A primeira API é um método para lidar com solicitações de páginas dinâmicas que foram disponível desde o início da web. É chamado de **CGI (Common Gate-interface de forma)** e é definido na RFC 3875. CGI fornece uma interface para permitir Servidores da Web para conversar com programas e scripts de back-end que podem aceitar entrada (por exemplo, de formulários) e gerar páginas HTML em resposta. Esses programas podem ser escrito em qualquer linguagem que seja conveniente para o desenvolvedor, geralmente um script

linguagem para facilitar o desenvolvimento. Escolha Python, Ruby, Perl ou sua LAN favorita calibre.

Por convenção, os programas invocados via CGI vivem em um diretório chamado *cgi-bin* , que é visível no URL. O servidor mapeia uma solicitação para este diretório para um nome do programa e executa esse programa como um processo separado. Ele fornece todos os dados enviado com a solicitação como entrada para o programa. A saída do programa dá um Página da Web que é retornada ao navegador.

Em nosso exemplo, o programa *order.cgi* é invocado com a entrada do formulário encodificado como mostrado na Fig. 7-26. Ele analisará os parâmetros e processará o pedido. UMA convenção útil é que o programa retornará o HTML para o formulário de pedido se nenhuma entrada de formulário é fornecida. Desta forma, o programa terá a certeza de conhecer o representação da forma.

A segunda API que veremos é bem diferente. A abordagem aqui é embutir pequenos scripts em páginas HTML e executá-los pelo servidor para gerar a página. Uma linguagem popular para escrever esses scripts é o **PHP** (**PHP: pré-processador de hipertexto**). Para usá-lo, o servidor precisa entender PHP, assim como um navegador precisa entender CSS para interpretar páginas da Web com folhas de estilo.

Normalmente, os servidores identificam páginas da Web que contêm PHP a partir da extensão de arquivo *php* em vez de *html* ou *htm* .

PHP é mais simples de usar do que CGI. Como exemplo de como funciona com formulários, veja o exemplo na Figura 7.30 (a). A parte superior desta figura contém um normal Página HTML com um formulário simples. Desta vez, a tag <form> especifica que *action.php* deve ser chamado para manipular os parâmetros quando o usuário envia o formulário. A página exibe duas caixas de texto, uma com uma solicitação de nome e outra com um pedido de uma idade. Depois que as duas caixas foram preenchidas e o formulário de envio feito, o servidor analisa a string do tipo Figura 7.26 enviada de volta, colocando o nome no variável *nome* e a idade na variável *idade* . Ele então começa a processar o *action.php* , mostrado na Figura 7.30 (b), como uma resposta. Durante o processamento deste arquivo,

Página 699

SEC. 7,3
THE WORLD WIDE WEB

675

os comandos PHP são executados. Se o usuário preencheu " Bárbara " e " 24 " no caixas, o arquivo HTML enviado de volta será o mostrado na Figura 7.30 (c). Assim, Criar formulários torna-se extremamente simples usando PHP.

```
<html>
<body>
<form action = "action.php" method = "post">
<p> Insira seu nome: <input type = "text" name = "name"></p>
<p> Insira sua idade: <input type = "text" name = "age"> </p>
<input type = "submit">
</form>
</body>
</html>
(uma)
<html>
<body>
<h1> Responder: </h1>
Olá <? Php echo $ name; ?>.
Previsão: no próximo ano você terá <? Php echo $ age + 1; ?>
</body>
</html>
(b)
<html>
<body>
<h1> Responder: </h1>
Hello Barbara.
```

Previsão: no próximo ano você fará 33

```
</body>
</html>
(c)
```

Figura 7-30. (a) Uma página da Web contendo um formulário. (b) Um script PHP para lidar com a saída do formulário. (c) Saída do script PHP quando as entradas são "Barbara" e "32", respectivamente.

Embora o PHP seja fácil de usar, na verdade é uma linguagem de programação poderosa para fazer a interface da Web e de um banco de dados do servidor. Possui variáveis, strings, matrizes e

a maioria das estruturas de controle encontradas em C, mas I / O muito mais poderoso do que apenas *printf*. PHP é um código-fonte aberto, disponível gratuitamente e amplamente utilizado. Foi desenhado especificamente para funcionar bem com o Apache, que também é de código aberto e é o servidor da Web mais usado do mundo. Para obter mais informações sobre PHP, consulte Valade (2009).

Agora vimos duas maneiras diferentes de gerar páginas HTML dinâmicas:

Scripts CGI e PHP embutido. Existem vários outros para escolher. **JSP** (**JavaServer Pages**) é semelhante ao PHP, exceto que a parte dinâmica é escrita em

Página 700

676

A CAMADA DE APLICAÇÃO

INDIVÍDUO. 7

a linguagem de programação Java em vez de PHP. Páginas que usam esta técnica tem a extensão de arquivo *.jsp* . **ASP.NET** (**Active Server Pages .NET**) é Micro-versão soft de PHP e JavaServer Pages. Ele usa programas escritos em Micro-estrutura de aplicativo em rede .NET proprietária da soft para gerar o dy- conteúdo dinâmico. As páginas que usam essa técnica têm a extensão *.aspx* . A escolha entre essas três técnicas geralmente tem mais a ver com política (código aberto vs. Microsoft) do que com a tecnologia, uma vez que as três linguagens são aproximadamente comparável.

Geração de página da web dinâmica do lado do cliente

Scripts PHP e CGI resolvem o problema de manipulação de entrada e interações com bancos de dados no servidor. Todos eles podem aceitar informações recebidas de formulários, procure informações em um ou mais bancos de dados e gere páginas HTML com os resultados. O que nenhum deles pode fazer é responder aos movimentos do mouse ou interagir com os usuários diretamente. Para isso, é necessário ter scripts em- alojado em páginas HTML que são executadas na máquina cliente em vez de máquina servidor. A partir do HTML 4.0, esses scripts são permitidos usando a tag *<script>* . As tecnologias usadas para produzir essas páginas da Web interativas são amplamente conhecido como **HTML dinâmico**

A linguagem de script mais popular para o lado do cliente é **JavaScript** , então nós irá agora dar uma olhada rápida nele. Apesar da semelhança nos nomes, o JavaScript tem quase nada a ver com a linguagem de programação Java. Como outros scripts linguagens, é uma linguagem de alto nível. Por exemplo, em uma única linha de JavaScript é possível abrir uma caixa de diálogo, aguardar a entrada de texto e armazenar o string resultante em uma variável. Recursos de alto nível como este tornam o JavaScript ideal para projetar páginas da Web interativas. Por outro lado, o fato de ser mutante andar mais rápido do que uma mosca de fruta presa em uma máquina de raios-X torna extremamente difícil escrever programas JavaScript que funcionem em todas as plataformas, mas talvez algum dia vai se estabilizar.

Como exemplo de um programa em JavaScript, considere o da Figura 7.31. Gostar aquele da Fig. 7-30, ele exibe um formulário pedindo um nome e idade, e então prevê quantos anos a pessoa terá no próximo ano. O corpo é quase o mesmo do PHP ex- amplo, a principal diferença é a declaração do botão *Enviar* e o declaração de atribuição nele. Esta declaração de atribuição diz ao navegador para invocar o script de *resposta* em um clique de botão e passa para ele o formulário como parâmetro.

O que é completamente novo aqui é a declaração da função JavaScript *response* no cabeçalho do arquivo HTML, uma área normalmente reservada para títulos, backcores de base e assim por diante. Esta função extrai o valor do campo de *nome* de o formulário e o armazena na variável *person* como uma string. Ele também extrai o valor do campo *idade*, converte em um número inteiro usando a função *eval*, adiciona 1 a ele, e armazena o resultado em *anos*. Em seguida, ele abre um documento para saída, faz quatro

Página 701

SEC. 7,3
THE WORLD WIDE WEB

677

```
<html>
<head>
<script language = "javascript" type = "text / javascript">
resposta da função (formulário de teste) {
var person = test form.name.value;
anos var = eval (formulário de teste.age.value) + 1;
document.open ();
document.writeln ("<html> <body>");
document.writeln ("Olá" + pessoa + ". <br>");
document.writeln ("Previsão: no próximo ano você terá" + anos + ".");
document.writeln ("</body> </html>");
document.close ();
}
</script>
</head>
<body>
<form>
Insira seu nome: <input type = "text" name = "name">
<p>
Digite sua idade: <input type = "text" name = "age">
<p>
<input type = "button" value = "submit" onclick = "response (this.form)">
</form>
</body>
</html>
```

Figura 7-31. Uso de JavaScript para processar um formulário.

grava nele usando o método *writeln* e fecha o documento. O documento é um arquivo HTML, como pode ser visto nas várias tags HTML nele. O navegador em seguida, exibe o documento na tela. É muito importante entender que, embora o PHP e o JavaScript sejam semelhantes na medida em que ambos incorporam código em arquivos HTML, eles são processados de forma totalmente diferente. No exemplo de PHP da Figura 7.30, depois que o usuário clica em *enviar* botão, o navegador coleta as informações em uma longa string e as envia para o servidor como um pedido de uma página PHP. O servidor carrega o arquivo PHP e executa o script PHP embutido para produzir uma nova página HTML. Aquela página é enviado de volta ao navegador para exibição. O navegador nem pode ter certeza de que foi produzido por um programa. Este processamento é mostrado nas etapas 1 a 4 na Fig. 7-32 (a).

No exemplo de JavaScript da Figura 7-31, quando o botão *enviar* é clicado, o navegador interpreta uma função JavaScript contida na página. Todo o trabalho é feito localmente, dentro do navegador. Não há contato com o servidor. Este processamento é mostrado nas etapas 1 e 2 na Figura 7-32 (b). Como consequência, o resultado é exibido virtualmente instantaneamente, enquanto com o PHP pode haver um atraso de sete vários segundos antes que o HTML resultante chegue ao cliente.

Página 702

678
A CAMADA DE APLICAÇÃO
INDIVÍDUO. 7
Servidor

```
Módulo PHP  
(uma)  
Navegador  
Do utilizador  
1  
4  
2  
3  
Servidor  
(b)  
Navegador  
Do utilizador  
1  
2  
JavaScript
```

Figura 7-32. (a) Script do lado do servidor com PHP. (b) Scripting do lado do cliente com JavaScript.

Essa diferença não significa que o JavaScript seja melhor do que o PHP. Seus usos são completamente diferentes. PHP (e, por implicação, JSP e ASP) é usado quando a interação com um banco de dados no servidor é necessária. JavaScript (e outras linguagens secundárias que iremos mencionar, como VBScript) é usado quando a interação é com o usuário no computador cliente. Certamente é possível combiná-los, pois veremos em breve.

JavaScript não é a única maneira de tornar as páginas da Web altamente interativas. Um alternativo nas plataformas Windows é o **VBScript**, que é baseado no Visual Basic.

Outro método popular entre plataformas é o uso de **miniaplicativos**. Estes são pequenos Programas Java que foram compilados em instruções de máquina para um sistema virtual computador chamado **JVM (Java Virtual Machine)**. Os miniaplicativos podem ser incorporados em

Páginas HTML (entre <applet> e </applet>) e interpretadas por JVM navegadores. Como os miniaplicativos Java são interpretados em vez de executados diretamente, o O interpretador Java pode impedi-los de fazer coisas ruins. Pelo menos em teoria. Na prática, os escritores de miniaplicativos encontraram um fluxo quase infinito de bugs no Java I / O bibliotecas para explorar.

A resposta da Microsoft aos applets Java da Sun foi permitir que as páginas da Web contivessem **Controles ActiveX**, que são programas compilados para linguagem de máquina x86 e executado no hardware básico. Esse recurso os torna muito mais rápidos e mais flexível do que miniaplicativos Java interpretados porque eles podem fazer qualquer coisa que um programa pode

Faz. Quando o Internet Explorer vê um controle ActiveX em uma página da Web, ele baixa ele, verifica sua identidade e o executa. No entanto, baixando e executando para- Os programas externos levantam enormes questões de segurança, que discutiremos no capítulo 8. Como quase todos os navegadores podem interpretar programas Java e JavaScript, um designer que deseja fazer uma página da Web altamente interativa tem uma escolha de pelo menos duas técnicas, e se a portabilidade para várias plataformas não for um problema, o ActiveX em Adição. Como regra geral, os programas JavaScript são mais fáceis de escrever, miniaplicativos Java execute mais rápido, e os controles ActiveX são executados mais rápido de todos. Além disso, uma vez que todos os navegadores implementar exatamente a mesma JVM, mas nenhum navegador implementa a mesma versão Com o JavaScript, os miniaplicativos Java são mais portáteis do que os programas JavaScript. Para mais informações sobre JavaScript, há muitos livros, cada um com muitos (frequentemente com mais de 1000) páginas. Veja, por exemplo, Flanagan (2010).

tecnologias-chave em uma combinação chamada **AJAX** (**A**synchronous **J**Avascript e **X**ml). Muitos aplicativos da Web com recursos completos, como Gmail, Maps e Docs, são escritos com AJAX.

AJAX é um pouco confuso porque não é uma linguagem. É um conjunto de tecnologia tecnologias que funcionam juntas para habilitar aplicativos da Web que são tão responsivos e poderosos como aplicativos de desktop tradicionais. As tecnologias são:

1. HTML e CSS para apresentar informações como páginas.
2. DOM (Document Object Model) para alterar partes das páginas enquanto são vistos.
3. XML (eXtensible Markup Language) para permitir a troca de programas de aplicação de dados com o servidor.
4. Uma maneira assíncrona para programas enviar e recuperar dados XML.
5. JavaScript como uma linguagem para unir todas essas funcionalidades.

Como esta é uma coleção e tanto, examinaremos cada peça para ver o que homenagens. Já vimos HTML e CSS. Eles são padrões para descrever conteúdo e como ele deve ser exibido. Qualquer programa que pode produzir HTML e CSS pode usar um navegador da Web como mecanismo de exibição.

DOM (**D**ocument **O**bject **M**odel) é uma representação de uma página HTML que está acessível a programas. Esta representação está estruturada como uma árvore que reflete a estrutura dos elementos HTML. Por exemplo, a árvore DOM do HTML em A Figura 7-30 (a) é apresentada na Figura 7-33. Na raiz está um elemento *html* que representa todo o bloco HTML. Este elemento é o pai do elemento *body* , que é por sua vez, pai de um elemento de *formulário* . O formulário tem dois atributos que são atraídos para o lado direito, um para o método do formulário (um *POST*) e outro para a ação do formulário (a URL a ser solicitada). Este elemento tem três filhos, refletindo os dois para-tags de gráfico e uma tag de entrada contidas no formulário. No fundo do a árvore são folhas que contêm elementos ou literais, como strings de texto. A importância do modelo DOM é que ele fornece programas com um maneira direta de alterar partes da página. Não há necessidade de reescrever a página inteira. Apenas o nó que contém a mudança precisa ser substituído. Quando esta alteração for feita, o navegador irá atualizar a tela de forma correspondente. Para exemplo, se uma imagem em parte da página for alterada no DOM, o navegador atualize essa imagem sem alterar as outras partes da página. Nós já temos visto o DOM em ação quando o exemplo de JavaScript da Figura 7.31 adicionou linhas ao

Página 704

680

A CAMADA DE APLICAÇÃO

INDIVÍDUO. 7

```
html
Atributos à direita
corpo
Formato
action = "action.php"
método = "postar"
p
p
"Por favor, insira
seu nome:"
tipo = "enviar"
entrada
entrada
tipo = "txt"
nome = "idade"
Elementos filho abaixo
Elementos
"Por favor, insira
sua idade:"
entrada
tipo = "txt"
nome = "idade"
```

Figura 7-33. A árvore DOM para o HTML na Figura 7.30 (a).

elemento do *documento* para fazer com que novas linhas de texto apareçam na parte inferior da sobrancelha ser janela. O DOM é um método poderoso para produzir páginas que podem evoluir. A terceira tecnologia, **XML (eXtensible Markup Language)**, é uma linguagem para especificar o conteúdo estruturado. HTML mistura conteúdo com formatação porque preocupa-se com a apresentação da informação. No entanto, como aplicativo da Web ções se tornam mais comuns, há uma necessidade crescente de separar conteúdo de sua apresentação. Por exemplo, considere um programa que pesquisa o Web pelo melhor preço de algum livro. Ele precisa analisar a aparência de muitas páginas da Web para o título e o preço do item. Com páginas da Web em HTML, é muito difícil para um programa descobrir onde está o título e onde está o preço. Por esta razão, o W3C desenvolveu XML (Bray et al., 2006) para permitir que a Web conteúdo a ser estruturado para processamento automatizado. Ao contrário do HTML, não há tags definidas para XML. Cada usuário pode definir suas próprias tags. Um exemplo simples de um documento XML é apresentado na Figura 7.34. Ele define uma estrutura chamada lista de livros , que é uma lista de livros . Cada livro tem três campos, o título, autor e ano de publicação. Essas estruturas são extremamente simples. É permitido ter estruturas turas com campos repetidos (por exemplo, vários autores), campos opcionais (por exemplo, URL do livro de áudio), e campos alternativos (por exemplo, URL de uma livraria se estiver impressa ou URL de um site de leilão, se estiver esgotado). Neste exemplo, cada um dos três campos é uma entidade indivisível, mas também é permitido subdividir ainda mais os campos. Por exemplo, o campo do autor pode ter feito da seguinte maneira para fornecer um controle mais refinado sobre a pesquisa e a formatação:

```
<author>
<nome> George </ nome>
<sobrenome> Zipf </ sobrenome>
</author>
```

Cada campo pode ser subdividido em subcampos e subcampos de forma arbitrária e profunda.

Página 705

SEC. 7,3
THE WORLD WIDE WEB

681

```
<? xml version = "1.0"?>
<lista de livros>
<book>
<title> Comportamento humano e o princípio do mínimo esforço </title>
<author> George Zipf </author>
<ano> 1949 </ano>
</book>
<book>
<title> A Teoria Matemática da Comunicação </title>
<author> Claude E. Shannon </author>
<author> Warren Weaver </author>
<ano> 1949 </ano>
</book>
<book>
<title> Mil novecentos e oitenta e quatro </title>
<author> George Orwell </author>
<ano> 1949 </ano>
</book>
</ lista de livros>
```

Figura 7-34. Um documento XML simples.

Tudo o que o arquivo da Figura 7.34 faz é definir uma lista de livros contendo três livros. Isto é bem adequado para transportar informações entre programas em execução em navegadores e servidores, mas não diz nada sobre como exibir o documento como uma página da web. Para fazer isso, um programa que consome as informações e julga 1949 para ser uma multa ano para livros pode gerar HTML em que os títulos são marcados como texto em itálico. Como alternativa, uma linguagem chamada **XSLT (eXtensible Stylesheet Language Transformações)**, pode ser usado para definir como o XML deve ser transformado em HTML. XSLT é como CSS, mas muito mais poderoso. Vamos poupar você dos detalhes.

A outra vantagem de expressar dados em XML, em vez de HTML, é que é mais fácil para os programas analisarem. HTML foi originalmente escrito manualmente (e frequentemente ainda está) então muito disso é um pouco desleixado. Às vezes, as tags de fechamento, como </p>, são deixadas. Fora, outras tags não têm uma tag de fechamento correspondente, como
. Ainda outras tags podem ser aninhados incorretamente, e a caixa dos nomes de tag e atributo pode variar. A maioria dos navegadores fazem o possível para descobrir o que provavelmente foi planejado. XML é mais rígido e mais limpo em sua definição. Nomes e atributos de tag são sempre em minúsculas, tags deve sempre ser fechado no reverso da ordem em que foram abertos (ou indique claramente se eles são uma tag vazia sem fechamento correspondente) e atribua os valores devem ser colocados entre aspas. Esta precisão torna a análise mais fácil e inequívoco.

HTML está até sendo definido em termos de XML. Esta abordagem é chamada **XHTML** (**eXtended HyperText Markup Language**). Basicamente, é muito

Página 706

682

A CAMADA DE APLICAÇÃO INDIVÍDUO. 7

Versão exigente de HTML. As páginas XHTML devem obedecer estritamente às regras XML, caso contrário, eles não são aceitos pelo navegador. Chega de páginas da web de má qualidade e inconsistências entre navegadores. Tal como acontece com XML, a intenção é produzir páginas que são melhores para programas (neste caso, aplicativos da Web) processar. Enquanto O XHTML existe desde 1998, mas demorou a pegar. Pessoas que produzem HTML, não vejo por que precisam de XHTML e o suporte ao navegador tem atrasado. Agora o HTML 5.0 está sendo definido para que uma página possa ser representada como ei-

o HTML ou XHTML para ajudar na transição. Eventualmente, o XHTML deve substituir HTML, mas vai demorar muito até que essa transição seja concluída.

XML também se provou popular como uma linguagem de comunicação entre gramas. Quando esta comunicação é realizada pelo protocolo HTTP (descrito em a próxima seção) é chamado de serviço da web. Em particular, **SOAP** (**S**imple **O**bject **P**rotocolo **d**e **A**cesso) é uma maneira de implementar serviços da Web que executa RPC para entre programas de forma independente de idioma e sistema. O cliente apenas con estrutura a solicitação como uma mensagem XML e a envia ao servidor, usando o método HTTP protocolo. O servidor envia de volta uma resposta como uma mensagem formatada em XML. Nisso forma, os aplicativos em plataformas heterogêneas podem se comunicar.

Voltando ao AJAX, nosso ponto é simplesmente que XML é um formato útil para ex alterar dados entre programas em execução no navegador e no servidor. Contudo, para fornecer uma interface responsiva no navegador ao enviar ou receber dados, deve ser possível para scripts realizar **E / S assíncronas** que não bloqueiam o exibir enquanto aguarda a resposta a uma solicitação. Por exemplo, considere um mapa que pode ser rolado no navegador. Quando é notificado da ação de rolagem, o script na página do mapa pode solicitar mais dados do mapa do servidor se a visualização de o mapa está próximo ao limite dos dados. A interface não deve congelar enquanto aqueles os dados são buscados. Essa interface não ganharia prêmios de usuário. Em vez disso, o scrolling deve continuar suavemente. Quando os dados chegam, o script é notificado para que ele pode usar os dados. Se tudo correr bem, novos dados de mapa serão buscados antes que sejam necessários

ed. Os navegadores modernos têm suporte para esse modelo de comunicação. A peça final do quebra-cabeça é uma linguagem de script que mantém o AJAX unido fornecendo acesso à lista de tecnologias acima. Na maioria dos casos, este idioma é JavaScript, mas existem alternativas, como VBScript. Apresentamos um simples exemplo de JavaScript anterior. Não se deixe enganar por essa simplicidade. JavaScript tem

muitas peculiaridades, mas é uma linguagem de programação desenvolvida, com todo o poder de C ou Java. Ele tem variáveis, strings, arrays, objetos, funções e todas as condições usuais estruturas trol. Ele também possui interfaces específicas para o navegador e páginas da web. JavaScript pode rastrear o movimento do mouse sobre os objetos na tela, o que o torna fácil fazer um menu aparecer repentinamente e levar a páginas da Web animadas. Pode usar DOM para acessar páginas, manipular HTML e XML e executar tarefas assíncronas

Comunicação HTTP.

Antes de deixar o assunto das páginas dinâmicas, vamos resumir brevemente as tecnologias que cobrimos até agora, relacionando-as em uma única figura. Com páginas completas da Web podem ser geradas instantaneamente por vários scripts no servidor

Página 707

SEC. 7,3
THE WORLD WIDE WEB

683

máquina. Os scripts podem ser escritos em linguagens de extensão de servidor como PHP, JSP, ou ASP.NET, ou executado como processos CGI separados e, portanto, ser escrito em qualquer idioma

calibre. Essas opções são mostradas na Figura 7-35.

Máquina servidor
CGI
roteiro
Ajudante
inscrição
Máquina cliente
Navegador da web
processo
PHP
ASP
JSP
Plug-ins
Java Script
intérprete
HTML / CSS /
Intérprete XML
Java virtual
máquina
Script VB
intérprete
XML
HTML / CSS
etc.
Navegador da web
processo

Figura 7-35. Várias tecnologias usadas para gerar páginas dinâmicas.

Assim que essas páginas da Web são recebidas pelo navegador, elas são tratadas como normais páginas em HTML, CSS e outros tipos de MIME e apenas exibidos. Plug-ins que funcionam no navegador e os aplicativos auxiliares executados fora do navegador podem estar em interrompido para estender os tipos MIME que são suportados pelo navegador.

A geração de conteúdo dinâmico também é possível no lado do cliente. Os programas que são incorporados em páginas da Web podem ser escritos em JavaScript, VBScript, Java e outras línguas. Esses programas podem realizar cálculos arbitrários e atualizar o display. Com AJAX, programas em páginas da Web podem trocar de forma assíncrona XML e outros tipos de dados com o servidor. Este modelo oferece suporte a aplicativos da Web ricos

cátions que se parecem com aplicativos tradicionais, exceto que são executados dentro do navegador e acessar informações armazenadas em servidores na Internet.

7.3.4 HTTP - o protocolo de transferência de hipertexto

Agora que entendemos o conteúdo e os aplicativos da Web, é hora de olhar para o protocolo que é usado para transportar todas essas informações entre Servidores e clientes da Web. É **HTTP (HyperText Transfer Protocol)**, como especificado citado na RFC 2616.

HTTP é um protocolo simples de solicitação-resposta que normalmente é executado sobre TCP. Isto especifica quais mensagens os clientes podem enviar aos servidores e quais respostas eles obtêm de volta em troca. Os cabeçalhos de solicitação e resposta são fornecidos em ASCII, assim como em

SMTP. O conteúdo é fornecido em um formato semelhante ao MIME, também como no SMTP. Isto modelo simples foi parcialmente responsável pelo sucesso inicial da Web porque tornou o desenvolvimento e a implantação simples.

Nesta seção, veremos as propriedades mais importantes do HTTP como ele é usado hoje em dia. No entanto, antes de entrar em detalhes, veremos que a maneira

Página 708

684

A CAMADA DE APLICAÇÃO
INDIVÍDUO. 7

ele é usado na Internet está evoluindo. HTTP é um protocolo de camada de aplicativo porque ele roda em cima do TCP e está intimamente associado à web. É por isso que estamos cobrindo isso neste capítulo. No entanto, em outro sentido, o HTTP está se tornando mais como um protocolo de transporte que fornece uma maneira de os processos se comunicarem conteúdo além das fronteiras de diferentes redes. Esses processos não têm para ser um navegador da Web e um servidor da Web. Um media player pode usar HTTP para falar com um servidor e solicitar informações do álbum. O software antivírus pode usar HTTP para baixar as atualizações mais recentes. Os desenvolvedores podem usar HTTP para buscar arquivos de projeto.

Produtos eletrônicos de consumo, como porta-retratos digitais, costumam usar um dispositivo Servidor HTTP como interface para o mundo exterior. Comunicação máquina a máquina cada vez mais é executado em HTTP. Por exemplo, um servidor de linha aérea pode usar SOAP (um XML RPC sobre HTTP) para entrar em contato com um servidor de aluguel de automóveis e fazer um carro reserva, tudo como parte de um pacote de férias. Essas tendências provavelmente continuarão, junto com a expansão do uso de HTTP.

Conexões

A maneira usual de um navegador entrar em contato com um servidor é estabelecer um protocolo TCP conexão para a porta 80 na máquina do servidor, embora este procedimento não seja formalmente requeridos. O valor de usar TCP é que nem navegadores nem servidores precisam se preocupar com como lidar com mensagens longas, confiabilidade ou controle de congestionamento. Todos dessas questões são tratadas pela implementação do TCP.

No início da Web, com HTTP 1.0, depois que a conexão foi estabelecida um pecado A solicitação foi enviada e uma única resposta foi enviada de volta. Então o TCP conexão foi lançada. Em um mundo em que a página da Web típica consistia inteiramente de texto HTML, esse método era adequado. Rapidamente, a página da Web média cresceu para conter um grande número de links incorporados para conteúdo, como ícones e outros colírio para os olhos. Estabelecendo uma conexão TCP separada para transportar cada ícone único tornou-se uma forma muito cara de operar.

Essa observação levou ao HTTP 1.1, que oferece suporte a **conexões persistentes**. Com elas, é possível estabelecer uma conexão TCP, enviar uma solicitação e obter uma resposta e, em seguida, enviar solicitações adicionais e obter respostas adicionais. Isto estratégia também é chamada de **reutilização de conexão**. Ao amortizar a configuração do TCP, inicialização,

e liberar custos sobre várias solicitações, a sobrecarga relativa devido ao TCP é reproduzido por solicitação. Também é possível enviar solicitações de pipeline, ou seja, enviar solicitação 2

antes que a resposta à solicitação 1 tenha chegado.

A diferença de desempenho entre esses três casos é mostrada na Figura 7.36.

A parte (a) mostra três solicitações, uma após a outra e cada uma em uma connection. Vamos supor que isso represente uma página da Web com dois imagens no mesmo servidor. Os URLs das imagens são determinados como os principais a página é buscada, portanto, eles são buscados após a página principal. Hoje em dia, um típico página tem cerca de 40 outros objetos que devem ser buscados para apresentá-la, mas isso seria

tornar nossa figura muito grande, então usaremos apenas dois objetos embutidos.

Página 709

SEC. 7,3

THE WORLD WIDE WEB

685

(uma)

(b)

(c)

Pipelined
solicitações de
Configuração de conexão

HTTP

Resposta

HTTP

Solicitação

Configuração de conexão

Configuração de conexão

Tempo

Configuração de conexão

Configuração de conexão

Figura 7-36. HTTP com (a) conexões múltiplas e solicitações sequenciais. (b)

Uma conexão persistente e solicitações sequenciais. (c) Uma conexão persistente e solicitações em pipeline.

Na Figura 7.36 (b), a página é buscada com uma conexão persistente. Ou seja, o A conexão TCP é aberta no início, então as mesmas três solicitações são enviadas, um após o outro, como antes, e só então a conexão é fechada. Observe aquilo a busca é concluída mais rapidamente. Existem duas razões para a aceleração. Primeiro, não se perde tempo configurando conexões adicionais. Cada conexão TCP requer pelo menos um tempo de ida e volta para estabelecer. Em segundo lugar, a transferência do mesmo

as imagens avançam mais rapidamente. Por que é isso? É por causa do congestionamento do TCP trol. No início de uma conexão, o TCP usa o procedimento de inicialização lenta para aumentar o throughput até aprender o comportamento do caminho da rede. A consequência de este período de aquecimento é que várias conexões TCP curtas demoram desproporcionalmente mais tempo para transferir informações do que uma conexão TCP mais longa.

Finalmente, na Figura 7-36 (c), há uma conexão persistente e as solicitações são pipelined. Especificamente, a segunda e a terceira solicitações são enviadas em rápida sucessão assim que o suficiente da página principal for recuperado para identificar que as imagens deve ser buscado. As respostas para essas solicitações seguem eventualmente. Este método reduz o tempo de inatividade do servidor, melhorando ainda mais o desempenho.

No entanto, conexões persistentes não vêm de graça. Um novo problema que eles aumentar é quando fechar a conexão. Uma conexão com um servidor deve permanecer aberta enquanto a página carrega. O que então? Há uma boa chance de que o usuário clique em um link que solicita outra página do servidor. Se a conexão permanecer aberto, a próxima solicitação pode ser enviada imediatamente. No entanto, não há garantia que o cliente fará outra solicitação ao servidor em breve. Na prática,

Página 710

686

A CAMADA DE APLICAÇÃO

INDIVÍDUO. 7

clientes e servidores geralmente mantêm conexões persistentes abertas até que tenham sido ocioso por um curto período de tempo (por exemplo, 60 segundos) ou eles têm um grande número de conexões e precisa fechar alguns.

O leitor atento pode ter notado que há uma combinação que nós deixei de fora até agora. Também é possível enviar uma solicitação por conexão TCP, mas execute várias conexões TCP em paralelo. Este método de **conexão paralela** era amplamente utilizado por navegadores antes de conexões persistentes. Tem a mesma desvantagem como conexões sequenciais — overhead extra — mas desempenho muito melhor. Isso ocorre porque configurar e aumentar as conexões em paralelo esconde alguns

da latência. Em nosso exemplo, conexões para ambas as imagens incorporadas pode ser configurado ao mesmo tempo. No entanto, executando muitas conexões TCP para o mesmo servidor é desencorajado. A razão é que o TCP realiza controle de congestionamento para cada conexão de forma independente. Como consequência, as conexões competem uns contra os outros, causando perda adicional de pacotes e, em conjunto, são mais agressivos para os usuários da rede do que uma conexão individual. Conexões persistentes são superior e usado na preferência para conexões paralelas porque evitam a cabeça e não sofrem de problemas de congestionamento.

Métodos

Embora o HTTP tenha sido projetado para uso na Web, foi feito intencionalmente mais geral do que o necessário, com vistas a futuras utilizações orientadas a objetos. Por esta razão, operações, **métodos** chamados , além de apenas solicitar uma página da Web são suportados. Essa generalidade é o que permitiu que o SOAP viesse a existir.

Cada solicitação consiste em uma ou mais linhas de texto ASCII, com a primeira palavra sendo a primeira linha o nome do método solicitado. Os métodos integrados são listados na Figura 7-37. Os nomes diferenciam maiúsculas de minúsculas, então *GET* é permitido, mas não *get* .

Método

Descrição

PEGUE

Leia uma página da web

CABEÇA

Leia o cabeçalho de uma página da web

POSTAR

Anexar a uma página da web

COLOCAR

Armazene uma página da web

EXCLUIR

Remova a página da web

VESTÍGIO

Ecoar a solicitação recebida

CONECTAR

Conecte-se por meio de um proxy

OPÇÕES

Opções de consulta para uma página

Figura 7-37. Os métodos de solicitação HTTP integrados.

O método *GET* solicita que o servidor envie a página. (Quando dizemos " página " queremos dizer " objeto " no caso mais geral, mas pensando em uma página como o conteúdo)

Página 711

SEC. 7,3

THE WORLD WIDE WEB

687

de um arquivo é suficiente para compreender os conceitos.) A página está devidamente codificada em

MIME. A grande maioria das solicitações para servidores Web são *GETs* . A forma usual de *GET* é

GET o nome do arquivo HTTP / 1.1

onde o *nome do arquivo* nomeia a página a ser buscada e 1.1 é a versão do protocolo.

O método *HEAD* pede apenas o cabeçalho da mensagem, sem a página real.

Este método pode ser usado para coletar informações para fins de indexação ou apenas para teste a validade de um URL.

O método *POST* é usado quando os formulários são enviados. Tanto ele quanto *GET* são também usado para serviços da Web SOAP. Como *GET* , tem um URL, mas em vez de simplesmente recuperando uma página, ele carrega dados para o servidor (ou seja, o conteúdo do formulário ou Parâmetros RPC). O servidor então faz algo com os dados que dependem de uma URL, anexando conceitualmente os dados ao objeto. O efeito pode ser para comprar um item, por exemplo, ou chamar um procedimento. Finalmente, o método retorna uma página indicando o resultado.

Os métodos restantes não são muito usados para navegar na web. O *PUT*

método é o inverso de *GET* : em vez de ler a página, ele escreve a página. Isto

método torna possível construir uma coleção de páginas da Web em um servidor remoto. O corpo da solicitação contém a página. Pode ser codificado usando MIME, em caso em que as linhas após *PUT* podem incluir cabeçalhos de autenticação, para provar que o chamador realmente tem permissão para realizar a operação solicitada. *DELETE* faz o que você espera: remove a página, ou pelo menos indica que o servidor Web concordou em remover a página. Como com *PUT*, autenticação e permissão desempenham um papel importante aqui.

O método *TRACE* é para depuração. Ele instrui o servidor a enviar de volta o solicitação. Este método é útil quando as solicitações não estão sendo processadas corretamente e o cliente deseja saber qual solicitação o servidor realmente recebeu.

O método *CONNECT* permite que um usuário faça uma conexão a um servidor Web por meio de um dispositivo intermediário, como um cache da Web.

O método *OPTIONS* fornece uma maneira para o cliente consultar o servidor por um página e obter os métodos e cabeçalhos que podem ser usados com essa página.

Cada solicitação obtém uma resposta que consiste em uma linha de status e, possivelmente, informações adicionais (por exemplo, toda ou parte de uma página da Web). A linha de status contém um código de status de três dígitos informando se a solicitação foi atendida e, se não, por que não. O primeiro dígito é usado para dividir as respostas em cinco grupos principais, como mostrado na Fig. 7-38. Os códigos 1xx raramente são usados na prática. Os códigos 2xx significa que a solicitação foi tratada com sucesso e o conteúdo (se houver) está sendo retornou. Os códigos 3xx dizem ao cliente para procurar em outro lugar, usando um diferente URL ou em seu próprio cache (discutido posteriormente). Os códigos 4xx significam que a solicitação falhou devido a um erro do cliente, como uma solicitação inválida ou uma página inexistente. finalmente, os Erros 5xx significam que o próprio servidor tem um problema interno, seja devido a um erro no seu código ou para uma sobrecarga temporária.

Página 712

688

A CAMADA DE APLICAÇÃO

INDIVÍDUO. 7

Código

Significado

Exemplos

1xx

Em formação

100 = o servidor concorda em atender ao pedido do cliente

2xx

Sucesso

200 = solicitação bem-sucedida; 204 = nenhum conteúdo presente

3xx

Redirecionamento

301 = página movida; 304 = página em cache ainda válida

4xx

Erro do cliente

403 = página proibida; 404 = página não encontrada

5xx

Erro de servidor

500 = erro interno do servidor; 503 = tente novamente mais tarde

Figura 7-38. Os grupos de resposta do código de status.

Cabeçalhos de mensagens

A linha de solicitação (por exemplo, a linha com o método *GET*) pode ser seguida por ad-linhas adicionais com mais informações. Eles são chamados de **cabeçalhos de solicitação**. Isto as informações podem ser comparadas aos parâmetros de uma chamada de procedimento. Respostas também pode ter **cabeçalhos de resposta**. Alguns cabeçalhos podem ser usados em qualquer direção.

Uma seleção dos mais importantes é fornecida na Figura 7.39. Esta lista não é curta, então, como você pode imaginar, muitas vezes há uma variedade de cabeçalhos em cada solicitação e response.

O cabeçalho *User-Agent* permite que o cliente informe o servidor sobre sua implementação do ser (por exemplo, *Mozilla / 5.0* e *Chrome / 5.0.375.125*). Essa informação é útil para permitir que os servidores ajustem suas respostas ao navegador, uma vez que diferentes sers podem ter capacidades e comportamentos amplamente variados.

Os quatro cabeçalhos de *aceitação* dizem ao servidor o que o cliente está disposto a aceitar em o evento que tem um repertório limitado do que é aceitável. O primeiro cabeçalho especifica os tipos MIME que são bem-vindos (por exemplo, *text / html*). O segundo dá o conjunto de caracteres (por exemplo, *ISO-8859-5* ou *Unicode-1-1*). O terceiro trata da compressão métodos de ação (por exemplo, *gzip*). O quarto indica um idioma natural (por exemplo, *espanhol*). Se o servidor tiver uma escolha de páginas, ele pode usar essas informações para fornecer uma o cliente está procurando. Se não for capaz de satisfazer a solicitação, um código de erro é re-virou e a solicitação falha.

Os cabeçalhos *If-Modified-Since* e *If-None-Match* são usados com cache.

Eles permitem que o cliente solicite o envio de uma página apenas se a cópia em cache não for mais válido. Descreveremos o cache em breve.

O cabeçalho *Host* nomeia o servidor. É retirado do URL. Este cabeçalho é obrigatório. É usado porque alguns endereços IP podem servir a vários nomes DNS e o servidor precisa de alguma forma de dizer a qual host entregar a solicitação.

O cabeçalho de *autorização* é necessário para páginas protegidas. Nesse caso, o cliente pode ter que provar que tem o direito de ver a página solicitada. Este cabeçalho é usado para esse caso.

O cliente usa o cabeçalho *Referer* com erro ortográfico para fornecer a URL que se refere o URL que agora é solicitado. Na maioria das vezes, é o URL da página anterior.

Página 713

SEC. 7.3

THE WORLD WIDE WEB

689

Cabeçalho

Tipo

Conteúdo

Agente de usuário

Solicitação

Informações sobre o navegador e sua plataforma

Aceitar

Solicitação

O tipo de páginas que o cliente pode manipular

Accept-Charset

Solicitação

Os conjuntos de caracteres que são aceitáveis para o cliente

Aceitar-Codificação

Solicitação

As codificações de página que o cliente pode manipular

Accept-Language

Solicitação

As linguagens naturais que o cliente pode manipular

Se-Modificado-Desde

Solicitação

Hora e data para verificar a atualização

If-None-Match

Solicitação

Tags enviadas anteriormente para verificar a atualização

Hospedeiro

Solicitação

O nome DNS do servidor

Autorização

Solicitação

Uma lista das credenciais do cliente

Referer

Solicitação

O URL anterior de onde veio o pedido

Bolacha

Solicitação

Cookie definido anteriormente enviado de volta ao servidor

Set-Cookie
Resposta
Cookie para o cliente armazenar
Servidor
Resposta
Informação sobre o servidor
Codificação de conteúdo
Resposta
Como o conteúdo é codificado (por exemplo, *gzip*)
Content-Language
Resposta
A linguagem natural usada na página
Comprimento do conteúdo
Resposta
O comprimento da página em bytes
Tipo de conteúdo
Resposta
O tipo MIME da página
Content-Range
Resposta
Identifica uma parte do conteúdo da página
Última modificação
Resposta
Hora e data em que a página foi alterada pela última vez
Expira
Resposta
Hora e data em que a página deixa de ser válida
Localização
Resposta
Diz ao cliente para onde enviar o seu pedido
Intervalos de aceitação
Resposta
Indica que o servidor aceitará solicitações de intervalo de bytes
Encontro
Ambos
Data e hora em que a mensagem foi enviada
Alcance
Ambos
Identifica uma parte de uma página
Cache-Control
Ambos
Diretivas sobre como tratar caches
ETag
Ambos
Tag para o conteúdo da página
Melhoria
Ambos
O protocolo para o qual o remetente deseja mudar

Figura 7-39. Alguns cabeçalhos de mensagens HTTP.

Este cabeçalho é particularmente útil para rastrear a navegação na Web, pois informa aos servidores como um cliente chegou à página.

Embora os cookies sejam tratados no RFC 2109 em vez de no RFC 2616, eles também têm cabeçalhos. O cabeçalho *Set-Cookie* é como os servidores enviam cookies aos clientes. O espera-se que o cliente salve o cookie e o retorne em solicitações subsequentes ao servidor usando o cabeçalho *Cookie*. (Observe que há uma especificação mais recente para cookies com cabeçalhos mais recentes, RFC 2965, mas isso foi amplamente rejeitado por indústria e não é amplamente implementado.)

O cabeçalho *Last-Modified* informa quando a página foi modificada pela última vez, e o *Expires* informa por quanto tempo a página permanecerá válida. Ambos os cabeçalhos desempenham um papel importante no cache de página.

O cabeçalho *Location* é usado pelo servidor para informar ao cliente que ele deve tentar um URL diferente. Isso pode ser usado se a página foi movida ou para permitir vários URLs para se referir à mesma página (possivelmente em servidores diferentes). Também é usado para

empresas que têm uma página da Web principal no domínio *com*, mas redirecionam os clientes para um

página nacional ou regional com base em seus endereços IP ou idioma preferencial.

Se uma página for muito grande, um pequeno cliente pode não querer tudo de uma vez. Alguns servidores aceitarão solicitações de intervalos de bytes, de modo que a página pode ser buscada em várias pequenas unidades. O cabeçalho *Accept-Ranges* anuncia a vontade do servidor de lidar com esse tipo de solicitação de página parcial.

Agora chegamos aos cabeçalhos que podem ser usados em ambas as direções. The *Date* header pode ser usado em ambas as direções e contém a hora e a data em que a mensagem foi enviado, enquanto o cabeçalho *Range* informa o intervalo de bytes da página que é fornecido por a resposta.

O cabeçalho *ETag* fornece uma pequena tag que serve como um nome para o conteúdo da página. É usado para armazenamento em cache. O cabeçalho *Cache-Control* fornece outras informações

instruções sobre como armazenar em cache (ou, mais comumente, como não armazenar em cache) páginas.

Finalmente, o cabeçalho de *atualização* é usado para mudar para uma nova comunicação protocolo, como um protocolo HTTP futuro ou um transporte seguro. Permite ao cliente para anunciar o que pode suportar e o servidor para afirmar o que está usando.

Cache

As pessoas costumam retornar a páginas da web que visualizaram antes e relacionadas

As páginas da Web geralmente têm os mesmos recursos incorporados. Alguns exemplos são as imagens que são usadas para navegação no site, bem como folhas de estilo comuns e scripts. Seria um grande desperdício buscar todos esses recursos para essas páginas cada vez que são exibidas porque o navegador já tem uma cópia.

A retirada de páginas buscadas para uso subsequente é chamada de **cache**.

A vantagem é que quando uma página em cache pode ser reutilizada, não é necessário turbar a transferência. HTTP tem suporte embutido para ajudar os clientes a identificar quando podem reutilizar páginas com segurança. Este suporte melhora o desempenho reduzindo ambos os tráfego e latência. A desvantagem é que o navegador agora deve armazenar páginas, mas isso quase sempre é uma troca que vale a pena, porque o armazenamento local é barato.

As páginas geralmente são mantidas no disco para que possam ser usadas quando o navegador for executado em uma data posterior.

O problema difícil com o cache HTTP é como determinar que uma cópia em cache de uma página é a mesma que a página seria se fosse buscada novamente.

Essa determinação não pode ser feita apenas a partir do URL. Por exemplo, o URL pode fornecer uma página que exibe as últimas notícias. O conteúdo desta página irá ser atualizado com frequência, embora o URL permaneça o mesmo. Alternativamente, o conteúdo da página pode ser uma lista dos deuses da mitologia grega e romana.

Esta página deve mudar um pouco menos rapidamente.

O HTTP usa duas estratégias para resolver esse problema. Eles são mostrados na Fig. 7-40 como formas de processamento entre a solicitação (etapa 1) e a resposta (etapa 5). A primeira estratégia é a validação da página (etapa 2). O cache é consultado, e se houver uma cópia de uma página para o URL solicitado que seja recente (ou seja, ainda válida), não há necessidade de buscá-lo novamente no servidor. Em vez disso, a página em cache pode ser

retornou diretamente. O cabeçalho *Expires* retornado quando a página em cache foi originada finalmente buscados e a data e hora atuais podem ser usados para fazer esta determinação ção.

4a: Não modificado
Navegador da web
Cache
servidor web
2: Verifique a validade
1: Pedido
3: GET condicional
4b: Resposta
5: Resposta
Programa

Figura 7-40. Cache HTTP.

No entanto, nem todas as páginas vêm com um conveniente cabeçalho *Expires* que informa quando a página deve ser buscada novamente. Afinal, fazer previsões é difícil, especialmente sobre o futuro. Nesse caso, o navegador pode usar heurísticas. Por exemplo, se o a página não foi modificada no ano passado (conforme informado pelo cabeçalho *Last-Modified*) é uma aposta bastante segura de que não mudará na próxima hora. Não há garantia tee, no entanto, e esta pode ser uma aposta ruim. Por exemplo, o mercado de ações pode fechou para o dia para que a página não mude por horas, mas vai mudar rapidamente assim que a próxima sessão de negociação começar. Assim, a capacidade de cache de um

a página pode variar muito com o tempo. Por este motivo, heurísticas devem ser usadas com cuidado, embora muitas vezes funcionem bem na prática.

Encontrar páginas que não expiraram é o uso mais benéfico do armazenamento em cache. porque isso significa que o servidor não precisa ser contatado. Infelizmente, nem sempre funciona. Os servidores devem usar o cabeçalho *Expires* de forma conservadora, já que eles podem não ter certeza de quando uma página será atualizada. Assim, as cópias em cache ainda pode estar fresco, mas o cliente não sabe.

A segunda estratégia é usada neste caso. É perguntar ao servidor se o cache a cópia ainda é válida. Essa solicitação é um **GET condicional** e é mostrada na Figura 7.40 como etapa 3. Se o servidor souber que a cópia em cache ainda é válida, ele pode enviar um breve responda para dizer isso (etapa 4a). Caso contrário, deve enviar a resposta completa (etapa 4b).

Página 716

692

A CAMADA DE APLICAÇÃO

INDIVÍDUO. 7

Mais campos de cabeçalho são usados para permitir que o servidor verifique se uma cópia em cache é

ainda válido. O cliente tem o tempo que uma página em cache foi atualizado a partir do *de último Cabeçalho modificado*. Ele pode enviar esse tempo para o servidor usando o *If-Modified-Since* cabeçalho para solicitar a página apenas se ela tiver sido alterada entretanto.

Como alternativa, o servidor pode retornar um cabeçalho *ETag* com uma página. Este cabeçalho dá uma tag que é um nome curto para o conteúdo da página, como uma soma de verificação, mas Melhor. (Pode ser um hash criptográfico, que descreveremos no Capítulo 8.) O cliente pode validar cópias em cache enviando ao servidor um cabeçalho *If-None-Match* listando as tags das cópias em cache. Se alguma das tags corresponder ao conteúdo que o servidor responderia com, a cópia em cache correspondente pode ser usada. Isto

O método pode ser usado quando não for conveniente ou útil determinar o frescor.

Por exemplo, um servidor pode retornar conteúdo diferente para o mesmo URL, dependendo em quais idiomas e tipos de MIME são preferidos. Neste caso, a modificação a data por si só não ajudará o servidor a determinar se a página em cache é nova.

Finalmente, observe que ambas as estratégias de cache são substituídas pela direc-tives carregados no cabeçalho *Cache-Control*. Essas diretivas podem ser usadas para restringir cache (por exemplo, *sem cache*) quando não for apropriado. Um exemplo é um dinâmico página que será diferente na próxima vez que for buscada. Páginas que requerem autorização também não são armazenados em cache.

Há muito mais coisas sobre o cache, mas só temos espaço para fazer dois im-

pontos importantes. Primeiro, o armazenamento em cache pode ser realizado em outros lugares além do

navegador. No caso geral, as solicitações HTTP podem ser encaminhadas por meio de uma série de caches. O uso de um cache externo ao navegador é denominado **cache de proxy**. Cada nível adicional de armazenamento em cache pode ajudar a reduzir as solicitações mais acima na cadeia. É com-

mon para organizações como ISPs e empresas para executar caches de proxy para obter os benefícios de armazenar páginas em cache em diferentes usuários. Discutiremos o cache de proxy com o tópico mais amplo de distribuição de conteúdo na Sec. 7.5 no final deste capítulo.

Em segundo lugar, os caches fornecem um impulso importante para o desempenho, mas não tanto quanto

pode-se esperar. A razão é que, embora existam documentos populares na Web, também há muitos documentos impopulares que as pessoas buscam, muitos dos quais também são muito longos (por exemplo, vídeos). A "cauda longa" de documentos impopulares

ocupam espaço em caches e o número de solicitações que podem ser tratadas do cache cresce lentamente com o tamanho do cache. Caches da web são todos maneiras mais prováveis de lidar com menos da metade das solicitações. Ver Breslau et al. (1999) para mais informações.

Experimentando com HTTP

Como o HTTP é um protocolo ASCII, é muito fácil para uma pessoa em um terminal (ao contrário de um navegador) para falar diretamente com os servidores da web. Tudo o que é necessário é um

Conexão TCP para a porta 80 no servidor. Os leitores são incentivados a experimentar com a seguinte sequência de comando. Funcionará na maioria dos shells UNIX e no janela de comando no Windows (uma vez que o programa telnet esteja ativado).

Página 717

SEC. 7,3

THE WORLD WIDE WEB

693

telnet www.ietf.org 80

GET /rfc.html HTTP / 1.1

Host: www.ietf.org

Esta sequência de comandos inicia uma conexão telnet (ou seja, TCP) para a porta 80 em Servidor da Web da IETF, www.ietf.org. Em seguida, vem o comando *GET* nomeando o caminho do URL e do protocolo. Experimente servidores e URLs de sua escolha. A próxima linha é o cabeçalho obrigatório do *Host*. Uma linha em branco após o último cabeçalho é obrigatório. Ele informa ao servidor que não há mais cabeçalhos de solicitação. O servidor irá então enviar a resposta. Dependendo do servidor e do URL, muitos tipos de cabeçalhos e páginas podem ser observados.

7.3.5 A Web Móvel

A Web é usada em quase todos os tipos de computador, e isso inclui dispositivos móveis telefones. Navegar na web em uma rede sem fio enquanto o celular pode ser muito útil - ful. Também apresenta problemas técnicos porque muito conteúdo da Web foi projetado para apresentações chamativas em computadores desktop com conectividade de banda larga. No Nesta seção, vamos descrever como acesso à Web a partir de dispositivos móveis, ou o **móvel Web**, está sendo desenvolvido.

Em comparação com computadores desktop no trabalho ou em casa, os telefones celulares estão presentes

várias dificuldades para navegar na Web:

1. Telas relativamente pequenas impedem páginas grandes e imagens grandes.
2. Recursos de entrada limitados tornam tedioso inserir URLs ou outros entradas longas.
3. A largura de banda da rede é limitada em links sem fio, especialmente em celulares redes lúlar (3G), onde geralmente é caro também.
4. A conectividade pode ser intermitente.

5. O poder de computação é limitado, por razões de vida útil da bateria, tamanho, calor dissipação e custo.

Essas dificuldades significam que simplesmente usar o conteúdo de desktop para a Web móvel é provavelmente proporcionará uma experiência de usuário frustrante.

As primeiras abordagens da web móvel criaram uma nova pilha de protocolos adaptada para dispositivos sem fio com recursos limitados. **WAP (Wireless Application Protocol)** é o exemplo mais conhecido dessa estratégia. O esforço WAP foi iniciado em 1997 por grandes fornecedores de telefones celulares, incluindo Nokia, Ericsson e Motorola. No entanto, algo inesperado aconteceu ao longo do caminho. Sobre o na próxima década, a largura de banda da rede e os recursos do dispositivo aumentaram enormemente com a implantação de serviços de dados 3G e telefones celulares com telas coloridas maiores,

Página 718

694

A CAMADA DE APLICAÇÃO INDIVÍDUO. 7

processadores mais rápidos e recursos sem fio 802.11. De repente, era possível para celulares rodarem navegadores da Web simples. Ainda há uma lacuna entre estes celulares e desktops que nunca fecham, mas muitos dos problemas de tecnologia que deu ímpeto a uma pilha de protocolo separada desapareceram.

A abordagem que é cada vez mais usada é executar os mesmos protocolos da Web para celulares e desktops, e ter sites que entreguem conteúdo otimizado para celular quando o usuário está em um dispositivo móvel. Os servidores da web são capazes de detectar se para retornar versões para desktop ou celular de páginas da Web, olhando para o cabeçalho de solicitação

ers. O cabeçalho *User-Agent* é especialmente útil a esse respeito porque identifica o software do navegador. Assim, quando um servidor Web recebe uma solicitação, ele pode olhar para os cabeçalhos e retornar uma página com imagens pequenas, menos texto e navegação mais simples para um iPhone e uma página cheia de recursos para um usuário em um laptop.

O W3C está incentivando essa abordagem de várias maneiras. Uma maneira é padronizar melhores práticas para conteúdo da Web móvel. Uma lista de 60 dessas práticas recomendadas é fornecida

na primeira especificação (Rabin e McCathieNevile, 2008). A maioria dessas práticas serviços tomam medidas sensatas para reduzir o tamanho das páginas, incluindo o uso de pressão, uma vez que os custos de comunicação são maiores do que os de computação, e maximizando a eficácia do armazenamento em cache. Essa abordagem incentiva sites, especialmente em sites grandes, para criar versões da Web para celular de seu conteúdo porque é tudo o que é necessário para capturar usuários da Web móvel. Para ajudar esses usuários, há também um logotipo para indicar as páginas que podem ser visualizadas (bem) na web móvel.

Outra ferramenta útil é uma versão simplificada de HTML chamada **XHTML Básico**. Esta linguagem é um subconjunto do XHTML que se destina ao uso em dispositivos móveis telefones, televisores, PDAs, máquinas de venda automática, pagers, carros, máquinas de jogos e até mesmo relógios. Por este motivo, ele não suporta folhas de estilo, scripts ou frames, mas a maioria das tags padrão está lá. Eles são agrupados em 11 módulos. Alguns é requerido; alguns são opcionais. Todos são definidos em XML. Os módulos e alguns tags de exemplo estão listadas na Figura 7-41.

No entanto, nem todas as páginas serão projetadas para funcionar bem na web móvel.

Assim, uma abordagem complementar é o uso de **transformação de conteúdo** ou **transcodificação**. Nesta abordagem, um computador que fica entre o celular e o servidor recebe solicitações do celular, busca conteúdo do servidor e o transforma para conteúdo da Web móvel. Uma transformação simples é reduzir o tamanho de grandes imagens reformatando-as em uma resolução mais baixa. Muitos outros pequenos mas úteis transformações são possíveis. A transcodificação tem sido usada com algum sucesso desde os primeiros dias da Web móvel. Ver, por exemplo, Fox et al. (1996). Contudo, quando ambas as abordagens são usadas, há uma tensão entre o conteúdo móvel de-

cisões feitas pelo servidor e pelo transcodificador. Por exemplo, um site pode selecionar uma combinação específica de imagem e texto para um usuário da web móvel, apenas para que um transcodificador mude o formato da imagem. Nossa discussão até agora tem sido sobre conteúdo, não protocolos, pois é o conteúdo esse é o maior problema para compreender a Web móvel. No entanto, iremos brevemente mencionar a questão dos protocolos. Os protocolos HTTP, TCP e IP usados pelo

Página 719

SEC. 7,3
THE WORLD WIDE WEB

695

Módulo

Req.?

Função

Tags de exemplo

Estrutura

sim

Doc. estrutura

corpo, cabeça, html, título

Texto

sim

Em formação

br, código, dfn, em, h_n, kbd, p, forte

Hipertexto

sim

Hiperlinks

uma

Lista

sim

Listas detalhadas

dl, dt, dd, ol, ul, li

Formulários

Não

Preencher formulários

formulário, entrada, rótulo, opção, área de texto

Mesas

Não

Mesas retangulares

legenda, tabela, td, th, tr

Imagen

Não

As fotos

img

Objeto

Não

Applets, mapas, etc. objeto, param

Meta informação

Não

Informação extra

meta

Ligaçāo

Não

Semelhante a <a>

ligação

Base

Não

URL inicial

base

Figura 7-41. Os módulos e tags do XHTML Basic.

A Web pode consumir uma quantidade significativa de largura de banda em sobrecargas de protocolo, como

como cabeçalhos. Para resolver este problema, o WAP e outras soluções definidas especiais-pur-representam protocolos. Isso acaba sendo desnecessário. Tecnologia de compressão de cabeçalho tecnologias, como ROHC (RObust Header Compression) descrito no Cap. 6, pode reduzir as despesas gerais desses protocolos. Desta forma, é possível ter um conjunto

de protocolos (HTTP, TCP, IP) e usá-los em largura de banda alta ou baixa links. O uso em links de baixa largura de banda simplesmente requer que a compressão do cabeçalho ser ligado.

7.3.6 Pesquisa na web

Para terminar nossa descrição da Web, discutiremos o que é indiscutivelmente o aplicativo da Web de maior sucesso: pesquisa. Em 1998, Sergey Brin e Larry Page, em seguida, estudantes de graduação em Stanford, formaram uma startup chamada Google para construir uma aposta motor de busca na web. Eles estavam armados com a ideia então radical de que uma busca algoritmo que contou quantas vezes cada página foi apontada por outras páginas era uma medida melhor de sua importância do que quantas vezes continha a chave palavras sendo procuradas. Por exemplo, muitas páginas vinculam à página principal da Cisco, que torna esta página mais importante para um usuário pesquisando por "Cisco" do que uma página de saída-lado da empresa que usa a palavra "Cisco" muitas vezes.

Eles estavam certos. Foi possível construir um mecanismo de pesquisa melhor e as pessoas se aglomeraram nele. Apoiado por capital de risco, o Google cresceu tremendamente. isto tornou-se uma empresa pública em 2004, com uma capitalização de mercado de US \$ 23 bilhões. Por

2010, estima-se que execute mais de um milhão de servidores em data centers em todo o mundo.

Página 720

696

A CAMADA DE APLICAÇÃO INDIVÍDUO. 7

Em certo sentido, a pesquisa é simplesmente outro aplicativo da Web, embora um dos mais aplicativos da Web maduros, porque está em desenvolvimento desde o início dias da web. No entanto, a pesquisa na web provou ser indispensável no dia a dia uso. Estima-se que mais de um bilhão de pesquisas na web sejam feitas a cada dia. Pessoas procurando por todos os tipos de informação, use a pesquisa como ponto de partida. Por exemplo, para descobrir onde comprar Vegemite em Seattle, não há um site óbvio para usar como ponto de partida. Mas é provável que um mecanismo de pesquisa conheça uma página com o informações desejadas e podem direcioná-lo rapidamente para a resposta.

Para realizar uma pesquisa na web da maneira tradicional, o usuário direciona sua sobrancelha ser para a URL de um site de pesquisa na web. Os principais sites de busca incluem o Google, Yahoo! E Bing. Em seguida, o usuário envia os termos de pesquisa usando um formulário. Este ato faz com que o mecanismo de pesquisa execute uma consulta em seu banco de dados por páginas relevantes ou

imagens, ou qualquer tipo de recurso que está sendo pesquisado, e retorna o resultado como uma página dinâmica. O usuário pode então seguir os links para as páginas que foram encontradas. A pesquisa na web é um tópico interessante para discussão porque tem implicações para a concepção e utilização de redes. Primeiro, há a questão de como a pesquisa na web encontra páginas. O motor de busca da Web deve ter um banco de dados de páginas para executar um

inquerir. Cada página HTML pode conter links para outras páginas, e tudo entre esting (ou pelo menos pesquisável) está vinculado em algum lugar. Isso significa que é teórico É possível começar com um punhado de páginas e encontrar todas as outras páginas da web fazendo um percurso por todas as páginas e links. Esse processo é chamado de **rastreamento da web**.

Todos os motores de busca da web usam rastreadores da web.

Um problema com o rastreamento é o tipo de página que ele pode encontrar. Buscando estática documentos e seguir os links é fácil. No entanto, muitas páginas da Web contêm gramas que exibem páginas diferentes dependendo da interação do usuário. Um exemplo é um catálogo online para uma loja. O catálogo pode conter páginas dinâmicas criadas de um banco de dados de produtos e consultas para produtos diferentes. Este tipo de conteúdo é diferente das páginas estáticas que são fáceis de percorrer. Como os rastreadores da web encontram

essas páginas dinâmicas? A resposta é que, na maioria das vezes, não. Esse tipo de conteúdo oculto é chamado de **deep web**. Como pesquisar na web profunda é um problema aberto que os pesquisadores estão enfrentando agora. Ver, por exemplo, madhavan et al. (2008). Também existem convenções pelas quais os sites criam uma página (conhecido como *robots.txt*) para informar aos rastreadores quais partes dos sites devem ou não ser visitadas. Uma segunda consideração é como processar todos os dados rastreados. Deixar algoritmos de indexação sejam executados sobre a massa de dados, as páginas devem ser armazenadas. Estes companheiros variam, mas acredita-se que os principais motores de busca tenham um índice de dezenas de bilhões de páginas retiradas da parte visível da web. O tamanho médio da página é estimado em 320 KB. Esses números significam que uma cópia rastreada da Web leva na ordem de 20 petabytes ou 2×10^{16} bytes para armazenar. Embora este seja um verdadeiro grande número, é também uma quantidade de dados que podem ser armazenados e processados confortavelmente em centros de dados da Internet (Chang et al., 2006). Por exemplo, se os custos de armazenamento em disco \$ 20 / TB, então 2×10^4 TB custa \$ 400.000, o que não é exatamente uma grande quantia para empresas do tamanho do Google, Microsoft e Yahoo!. E enquanto a web é

Página 721

SEC. 7,3
THE WORLD WIDE WEB

697

em expansão, os custos de disco estão caindo drasticamente, então o armazenamento de toda a Web pode

continuar a ser viável para grandes empresas no futuro previsível.

Dar sentido a esses dados é outra questão. Você pode apreciar como o XML pode ajudar os programas a extrair a estrutura dos dados facilmente, enquanto os formatos ad hoc levará a muitas suposições. Há também a questão da conversão entre tapetes, e até tradução entre idiomas. Mas mesmo conhecendo a estrutura dos dados são apenas parte do problema. O difícil é entender o que isso significa. Isto é onde muito valor pode ser desbloqueado, começando com páginas de resultados mais relevantes para

consultas de pesquisa. O objetivo final é ser capaz de responder a perguntas, por exemplo, onde comprar uma torradeira barata, mas decente, em sua cidade.

Um terceiro aspecto da pesquisa na web é que ela passou a fornecer um nível mais alto de nomeação. Não há necessidade de lembrar um URL longo se for tão confiável (ou talvez mais) para pesquisar uma página da Web pelo nome de uma pessoa, supondo que você seja melhor em lembrar nomes do que URLs. Essa estratégia está cada vez mais bem-sucedida.

Da mesma forma que os nomes DNS relegaram os endereços IP aos computadores, a pesquisa está relegando URLs aos computadores. Também a favor da pesquisa é que ela corrige erros de ortografia e digitação, ao passo que, se você digitar um URL errado, obterá a página errada.

Finalmente, a pesquisa na web nos mostra algo que tem pouco a ver com a desinformação, mas muito a ver com o crescimento de alguns serviços de Internet: há muito dinheiro em publicidade. A publicidade é o motor econômico que impulsionou o crescimento da pesquisa na Web. A principal mudança da publicidade impressa é a capacidade de direcionar anúncios dependendo do que as pessoas estão procurando, para aumentar a relevância dos anúncios. Variações em um mecanismo de leilão são usadas para corresponder a consulta de pesquisa ao anúncio mais valioso (Edelman et al., 2007).

Este novo modelo deu origem a novos problemas, é claro, como a **fraude de cliques**, em quais programas imitam os usuários e clicam em anúncios para fazer pagamentos que não foram merecidos de forma justa.

7.4 TRANSMISSÃO DE ÁUDIO E VÍDEO

Os aplicativos da web e a web móvel não são os únicos desenvolvimentos interessantes no uso de redes. Para muitas pessoas, áudio e vídeo são o Santo Graal da rede

trabalhando. Quando a palavra "multimídia" é mencionada, ambos os propellerheads e os trajes começam a salivar como se estivessem na hora. O primeiro vê imensa técnica desafios em fornecer voz sobre IP e vídeo sob demanda para todos os computadores. Os últimos vêm lucros igualmente imensos nisso.

Embora a ideia de enviar áudio e vídeo pela Internet tenha existido desde os anos 1970, pelo menos, é apenas a partir de aproximadamente 2000 que o áudio e o tráfego de vídeo em tempo real cresceu intensamente. O tráfego em tempo real é diferente do tráfego da Web em que deve ser executado em uma taxa predeterminada para ser útil. Afinal, assistir a um vídeo em câmera lenta com intermitências não é

Página 722

698

A CAMADA DE APLICAÇÃO INDIVÍDUO. 7

a ideia de diversão das pessoas. Em contraste, a Web pode ter interrupções curtas e a página as cargas podem demorar mais ou menos tempo, dentro dos limites, sem ser um grande problema. Duas coisas aconteceram para possibilitar esse crescimento. Primeiro, os computadores se tornaram muito mais potentes e equipados com microfones e câmeras para que pode inserir, processar e produzir dados de áudio e vídeo com facilidade. Em segundo lugar, uma inundação de

A largura de banda da Internet passou a estar disponível. Links de longa distância no centro do Internet executado a muitos gigabits / s, e banda larga e 802.11 sem fio alcança os usuários na borda da Internet. Esses desenvolvimentos permitem que os ISPs carreguem um tremendo níveis de tráfego em seus backbones e significa que usuários comuns podem se conectar para a Internet 100–1000 vezes mais rápido do que com um modem telefônico de 56 kbps.

A inundação de largura de banda fez com que o tráfego de áudio e vídeo crescesse, mas para diferentes razões diferentes. As chamadas telefônicas ocupam relativamente pouca largura de banda (em princípio 64

kbps, mas menos quando comprimido), mas o serviço telefônico tradicionalmente foi expensivo. As empresas viram uma oportunidade de transportar tráfego de voz pela Internet usando a largura de banda existente para reduzir suas contas de telefone. Startups como O Skype viu uma maneira de permitir que os clientes façam ligações gratuitas usando a Internet conexões. Companhias telefônicas iniciantes viram uma maneira barata de transportar chamadas de voz usando equipamento de rede IP. O resultado foi uma explosão de voz dados transportados por redes da Internet que são chamados de **voz sobre IP** ou **Internet telefonia**.

Ao contrário do áudio, o vídeo ocupa uma grande quantidade de largura de banda. Razoável qualitativamente. O vídeo da Internet é codificado com compressão a taxas de cerca de 1 Mbps, e um filme em DVD típico tem 2 GB de dados. Antes do acesso à Internet banda larga, o envio de filmes na rede eram proibitivos. Não é mais assim. Com a propagação de banda larga, tornou-se possível, pela primeira vez, para os usuários assistirem vídeos decentes ed vídeo em casa. As pessoas adoram fazer isso. Cerca de um quarto dos usuários da Internet estimam-se que qualquer dia visite o YouTube, o popular site de compartilhamento de vídeos. O negócio de aluguel de filmes mudou para downloads online. E o tamanho do vídeo mudou a composição geral do tráfego da Internet. A maioria dos Internet tráfego da rede já é de vídeo, e estimam-se que 90% do tráfego da Internet será vídeo dentro de alguns anos (Cisco, 2010).

Dado que há largura de banda suficiente para transportar áudio e vídeo, a questão principal para projetar aplicativos de streaming e conferência é o atraso da rede. Áudio e o vídeo precisa de apresentação em tempo real, o que significa que deve ser reproduzido em uma taxa predeterminada para ser útil. Longos atrasos significam que chamadas que deveriam ser interativos não são mais. Este problema é claro se você já falou em um satélite telefone leve, onde o atraso de até meio segundo distrai bastante. Para jogar música e filmes pela rede, o atraso absoluto não importa, porque afeta apenas quando a mídia começa a ser reproduzida. Mas a variação no atraso, chamada **jitter**, ainda importa. Deve ser mascarado pelo reproduutor ou o áudio soará incorreto informativo e o vídeo parecerá irregular.

Nesta seção, discutiremos algumas estratégias para lidar com o problema de atraso, como bem como protocolos para configurar sessões de áudio e vídeo. Após uma introdução a

SEC. 7,4

TRANSMISSÃO DE ÁUDIO E VÍDEO

699

áudio e vídeo digital, nossa apresentação é dividida em três casos para os quais designs diferentes são usados. O primeiro e mais fácil caso de lidar é o streaming armazenado mídia, como assistir a um vídeo no YouTube. O próximo caso em termos de dificuldade é streaming de mídia ao vivo. Dois exemplos são rádio na Internet e IPTV, em que o rádio e estações de televisão transmitem para muitos usuários ao vivo pela Internet. O último e o caso mais difícil é uma chamada que pode ser feita com o Skype, ou mais geralmente um áudio e videoconferência interativa.

Como um aparte, o termo **multimídia** é frequentemente usado no contexto da Internet para significar vídeo e áudio. Literalmente, multimídia é apenas duas ou mais mídias. que definição torna este livro uma apresentação multimídia, pois contém texto e gráficos (as figuras). No entanto, provavelmente não é isso que você tinha em mente, então nós use o termo "multimídia" para implicar duas ou mais **mídias contínuas**, ou seja, mídia que deve ser reproduzida durante algum intervalo de tempo bem definido. Os dois a mídia normalmente é vídeo com áudio, ou seja, imagens em movimento com som. Muitos as pessoas também se referem a áudio puro, como telefonia na Internet ou rádio na Internet, como multimídia também, o que claramente não é. Na verdade, um termo melhor para todos esses casos é **streaming media**. No entanto, vamos seguir o rebanho e considerar áudio em tempo real para ser multimídia também.

7.4.1 Áudio Digital

Uma onda de áudio (som) é uma onda acústica (pressão) unidimensional. Quando uma onda acústica entra no ouvido, o tímpano vibra, causando os minúsculos ossos de o ouvido interno vibra junto com ele, enviando impulsos nervosos ao cérebro. Estes os pulsos são percebidos como som pelo ouvinte. De forma semelhante, quando um acústico onda atinge um microfone, o microfone gera um sinal elétrico, representando a amplitude do som em função do tempo.

A faixa de frequência do ouvido humano vai de 20 Hz a 20.000 Hz. Alguns animais, principalmente cães, podem ouvir frequências mais altas. O ouvido ouve o volume logarithmicamente, então a proporção de dois sons com potência A e B é convencionalmente expresso em **dB** (decibéis) como a quantidade $10 \log_{10} (A/B)$. Se definirmos o inferior limite de audibilidade (uma pressão sonora de cerca de $20 \mu\text{Pascais}$) para uma onda senoidal de 1 kHz

como 0 dB, uma conversa normal é de cerca de 50 dB e o limite de dor é de cerca 120 dB. A faixa dinâmica é um fator de mais de 1 milhão.

O ouvido é surpreendentemente sensível a variações de som que duram apenas alguns milissegundos. O olho, em contraste, não percebe mudanças no nível de luz que duram apenas alguns milissegundos. O resultado dessa observação é que o jitter de apenas alguns milissegundos durante a reprodução de multimídia afetam a qualidade de som percebida muito mais do que afeta a qualidade da imagem percebida.

O áudio digital é uma representação digital de uma onda de áudio que pode ser usada para recriá-lo. As ondas de áudio podem ser convertidas para a forma digital por um **ADC** (Analog-conversor para digital). Um ADC leva uma tensão elétrica como entrada e gera um número binário como saída. Na Figura 7.42 (a), vemos um exemplo de onda senoidal.

700

A CAMADA DE APLICAÇÃO
INDIVÍDUO. 7

Para representar este sinal digitalmente, podemos amostrá-lo a cada ΔT segundos, conforme mostrado por

as alturas das barras na Figura 7.42 (b). Se uma onda sonora não é uma onda senoidal pura, mas um superposição linear de ondas senoidais onde o componente de maior frequência está presente é f , o teorema de Nyquist (ver Capítulo 2) afirma que é suficiente fazer amostras em uma frequência $2f$. A amostragem com mais frequência não tem valor, pois as frequências mais altas que tal amostragem poderia detectar não estão presentes.

1,00
0,75
0,50
0,25
0
-0,25
-0,50
-0,75
-1,00

1
2
T
1
2
T
T
T
(uma)
(b)
(c)

Figura 7-42. (a) Uma onda senoidal. (b) Amostragem da onda senoidal. (c) Quantizando o amostras para 4 bits.

O processo reverso assume valores digitais e produz um sistema elétrico analógico Voltagem. Isso é feito por um **DAC** (**conversor digital para analógico**). Um alto-falante pode em seguida, converter a voltagem analógica em ondas acústicas para que as pessoas possam ouvir os sons.

Amostras digitais nunca são exatas. As amostras da Fig. 7-42 (c) permitem apenas nove valores, de -1,00 a +1,00 em etapas de 0,25. Uma amostra de 8 bits permitiria 256 valores distintos. Uma amostra de 16 bits permitiria 65.536 valores distintos. O erro em introduzido pelo número finito de bits por amostra é chamado de **ruído de quantização**. Se for muito grande, o ouvido o detecta.

Dois exemplos bem conhecidos em que o som amostrado é usado são o telefone e discos compactos de áudio. Modulação por código de pulso, conforme usado no telefone sistema, usa amostras de 8 bits feitas 8.000 vezes por segundo. A escala é não linear para minimizar a distorção percebida, e com apenas 8.000 amostras / s, frequências acima 4 kHz são perdidos. Na América do Norte e no Japão, a codificação μ -law é usada. No Europa e internacionalmente, a codificação A-law é usada. Cada codificação oferece um taxa de dados de 64.000 bps.

Os CDs de áudio são digitais com uma taxa de amostragem de 44.100 amostras / s, o suficiente para capturar frequências de até 22.050 Hz, o que é bom o suficiente para as pessoas, mas ruim para amantes da música canina. As amostras têm 16 bits cada e são lineares no intervalo de amplitudes. Observe que as amostras de 16 bits permitem apenas 65.536 valores distintos, mesmo embora a faixa dinâmica do ouvido seja superior a 1 milhão. Assim, embora O áudio com qualidade de CD é muito melhor do que o áudio com qualidade de telefone, usando apenas 16 bits por amostra introduz ruído de quantização perceptível (embora toda a dinâmica o intervalo não é coberto - os CDs não devem doer). Alguns audiófilos fanáticos

arranhões, a menos que sejam manuseados com muito cuidado) Com 44.100 amostras / s de 16 bits cada, áudio com qualidade de CD não compactado precisa de uma largura de banda de 705,6 kbps para mono e 1,411 Mbps para estéreo.

Compressão de Áudio

O áudio é frequentemente compactado para reduzir as necessidades de largura de banda e tempos de transferência, até mesmo embora as taxas de dados de áudio sejam muito mais baixas do que as taxas de dados de vídeo. Toda compressão sistemas requerem dois algoritmos: um para compactar os dados na fonte, e outro para descompactá-lo no destino. Na literatura, esses algoritmos são chamados de **codificação e decodificação**, respectivamente. Nós vamos usar essa terminologia também.

Os algoritmos de compressão exibem certas assimetrias que são importantes para desarmar o entendimento. Mesmo que estejamos considerando o áudio primeiro, essas assimetrias são válidas para vídeo também. Para muitas aplicações, um documento multimídia só será inserido codificado uma vez (quando é armazenado no servidor multimídia), mas será decodificado muitas vezes (quando é reproduzido pelos clientes). Essa assimetria significa que é aceitável que o algoritmo de codificação seja lento e requeira dispêndios de software, desde que o algoritmo de decodificação seja rápido e não requeira caro hardware. O operador de um servidor de áudio (ou vídeo) popular pode ser bastante curioso tentando comprar um cluster de computadores para codificar toda a sua biblioteca, mas exigindo usuários que façam o mesmo para ouvir música ou assistir a filmes provavelmente não será um grande sucesso.

Muitos sistemas de compressão práticos vão longe para fazer a decodificação de forma rápida e simples, mesmo ao preço de tornar a codificação lenta e complicada. Por outro lado, para áudio e vídeo ao vivo, como chamadas de voz sobre IP, a codificação lenta é inaceitável. A codificação deve acontecer em tempo real, em tempo real. Consequentemente, a multimídia em tempo real usa algoritmos ou parâmetros diferentes dos áudio ou vídeos armazenados no disco, muitas vezes com compressão consideravelmente menor. Uma segunda assimetria é que o processo de codificação / decodificação não precisa ser invertível. Ou seja, ao compactar um arquivo de dados, transmiti-lo e, em seguida, descompactá-lo, o usuário espera obter o original de volta, com precisão até o último bit. Com multimídia, este requisito não existe. Geralmente é aceitável ter o áudio (ou vídeo) sinal após a codificação e decodificação ser ligeiramente diferente do original, desde que seja (ou pareça) o mesmo. Quando a saída decodificada não é exatamente igual à entrada original, o sistema é considerado com **perdas**. Se a entrada e saída são idênticos, o sistema é **sem perdas**. Sistemas com perdas são importantes porque aceitar uma pequena quantidade de perda de informações normalmente significa uma grande recompensa em termos da taxa de compressão possível.

Historicamente, a largura de banda de longa distância na rede telefônica era muito cara, portanto, há um corpo substancial de trabalho sobre **vocoders** (abreviação de "codificadores de voz") que compactam áudio para o caso especial da fala. A fala humana tende a ser

a faixa de 600 Hz a 6000 Hz e é produzida por um processo mecânico que depende no trato vocal, língua e mandíbula do falante. Alguns vocoders fazem uso de modelos do sistema vocal para reduzir a fala a alguns parâmetros (por exemplo, os tamanhos e formatos de várias cavidades) e uma taxa de dados de apenas 2,4 kbps. Como estes o trabalho dos codificadores de voz está além do escopo deste livro, entretanto.

Vamos nos concentrar no áudio enviado pela Internet, que normalmente é

mais perto da qualidade de CD. Também é desejável reduzir as taxas de dados para este tipo de áudio. A 1,411 Mbps, o áudio estéreo obstruiria muitos links de banda larga, deixando menos espaço para vídeo e outro tráfego da web. Sua taxa de dados com compressão pode ser reduzido em uma ordem de magnitude com pouca ou nenhuma perda percebida de qualidade. A compressão e a descompressão requerem processamento de sinal. Felizmente, digisom e filmes otimizados podem ser facilmente processados por computadores em software. De fato, dezenas de programas existem para permitir aos usuários gravar, exibir, editar, misturar e armazenar mídia

de várias fontes. Isso levou a uma grande quantidade de músicas e filmes sendo disponível na Internet - nem tudo legal - o que resultou em várias leis processos dos artistas e proprietários de direitos autorais.

Muitos algoritmos de compressão de áudio foram desenvolvidos. Provavelmente o mais formatos populares são **MP3** (**camada de áudio MPEG 3**) e **AAC** (**áudio avançado Codificação**) conforme transportado em **arquivos MP4** (**MPEG-4**). Para evitar confusão, observe que

MPEG fornece compressão de áudio e vídeo. MP3 refere-se à compressão de áudio parte da seção (parte 3) do padrão MPEG-1, não a terceira versão do MPEG. Na verdade, nenhuma terceira versão do MPEG foi lançada, apenas MPEG-1, MPEG-2 e MPEG-4. AAC é o sucessor do MP3 e a codificação de áudio padrão usada em MPEG-4. MPEG-2 permite áudio MP3 e AAC. Isso está claro agora? O bom

O problema dos padrões é que existem muitos para escolher. E se você não

como qualquer um deles, espere um ou dois anos.

A compressão de áudio pode ser feita de duas maneiras. Na **codificação da forma de onda**, o sinal é transformado matematicamente por uma transformada de Fourier em sua frequência componentes. No cap. 2, mostramos uma função de exemplo de tempo e seu Fourier amplitudes na Figura 2.1 (a). A amplitude de cada componente é então codificada em um maneira mínima. O objetivo é reproduzir a forma de onda com bastante precisão no outra extremidade com o mínimo de bits possível.

A outra forma, a **codificação perceptual**, explora certas falhas no sistema de áudio humano sistema histórico para codificar um sinal de tal forma que soe igual para um humano ouvinte, mesmo que pareça bem diferente em um osciloscópio. A codificação perceptiva é com base na ciência da **psicoacústica** - como as pessoas percebem o som. Ambos MP3 e AAC são baseados na codificação perceptual.

A propriedade chave da codificação perceptual é que alguns sons podem **mascarar** outros sons. Imagine que você está transmitindo um concerto de flauta ao vivo em um dia quente de verão. Então, de repente, do nada, uma equipe de trabalhadores nas proximidades liga suas britadeiras e começar a rasgar a rua. Ninguém mais consegue ouvir a flauta. Seus sons foram mascarados pelas britadeiras. Para fins de transmissão, agora é suficiente para codificar apenas a banda de frequência usada pelas britadeiras

Página 727

SEC. 7,4 TRANSMISSÃO DE ÁUDIO E VÍDEO

703

porque os ouvintes não podem ouvir a flauta de qualquer maneira. Isso é chamado de **frequência mascaramento** - a capacidade de um som alto em uma banda de frequência para ocultar um som mais suave

em outra banda de frequência que teria sido audível na ausência do som som. Na verdade, mesmo depois que as britadeiras pararem, a flauta será inaudível por um curto período de tempo porque o ouvido diminui seu ganho quando eles começam e leva um tempo finito para aumentá-lo novamente. Este efeito é chamado de **mascaramento temporal**.

Para tornar esses efeitos mais quantitativos, imagine o experimento 1. Uma pessoa em um sala silenciosa coloca fones de ouvido conectados à placa de som de um computador. O computador gera uma onda senoidal pura em 100 Hz em baixa, mas aumentando gradualmente, a potência

er. A pessoa é instruída a apertar uma tecla ao ouvir o tom. O com

computador registra o nível de potência atual e, em seguida, repete o experimento a 200 Hz, 300 Hz, e todas as outras frequências até o limite da audição humana. Quando em média de muitas pessoas, um gráfico log-log de quanta energia é necessária para um o tom a ser audível é semelhante ao da Figura 7-43 (a). Uma consequência direta disso curva é que nunca é necessário codificar quaisquer frequências cuja potência cai abaixo do limiar de audibilidade. Por exemplo, se a potência em 100 Hz fosse 20 dB na Fig. 7-43 (a), ele poderia ser omitido da saída sem perda perceptível de qualidade porque 20 dB a 100 Hz fica abaixo do nível de audibilidade.

Mascaramento

sinal em

150 Hz

Límite

de audibilidade

80

60

40

20

.1

Frequência (kHz)

P

o

W

e

r

(dB)

10

20

5

0,5

0,05

0,2

2

0

80

60

40

20

Frequência (kHz)

(uma)

(b)

Mascarado

sinal

Límite

de audibilidade

P

o

W

e

r

(dB)

0

.1

1

10

20

5

0,5

0,05

0,2

2

0,02

2

Figura 7-43. (a) O limite de audibilidade em função da frequência. (b) O efeito de mascaramento.

Agora considere o experimento 2. O computador executa o experimento 1 novamente, mas este tempo com uma onda senoidal de amplitude constante em, digamos, 150 Hz sobreposta no teste frequência. O que descobrimos é que o limite de audibilidade para frequências próximo a 150 Hz é elevado, como mostra a Figura 7.43 (b).

A consequência desta nova observação é que, mantendo o controle de quais sinais estão sendo mascarados por sinais mais poderosos em bandas de frequência próximas, nós pode omitir mais e mais frequências no sinal codificado, economizando bits. Na Fig. 7-43, o sinal de 125 Hz pode ser completamente omitido da saída e ninguém ser capaz de ouvir a diferença. Mesmo depois de um sinal poderoso parar em algum tempo banda de frequência, o conhecimento de suas propriedades de mascaramento temporal nos permite continuar omitir as frequências mascaradas por algum intervalo de tempo enquanto o ouvido se recupera. o

A CAMADA DE APLICAÇÃO INDIVÍDUO. 7

A essência do MP3 e AAC é transformar o som de Fourier para obter o poder de cada frequência e, em seguida, transmitir apenas as frequências não mascaradas, codificando-as em tão poucos bits quanto possível.

Com essas informações como pano de fundo, agora podemos ver como a codificação está feito. A compressão de áudio é feita amostrando a forma de onda a uma taxa de 8 a 96 kHz para AAC, geralmente em 44,1 kHz, para imitar o som de CD. A amostragem pode ser feito em um (mono) ou dois canais (estéreo). Em seguida, a taxa de bits de saída é escolhida. MP3 pode comprimir um CD de rock 'n roll estéreo até 96 kbps com pouca percepção perda de qualidade, mesmo para fãs de rock 'n roll sem perda auditiva. Para um piano com cert, AAC com pelo menos 128 kbps é necessário. A diferença é porque o sinal a taxa de ruído para o rock 'n roll é muito maior do que para um concerto de piano (em um sentido estranho, de qualquer maneira). Também é possível escolher taxas de produção mais baixas e aceitar

alguma perda de qualidade.

As amostras são processadas em pequenos lotes. Cada lote é passado por um banco de filtros digitais para obter bandas de frequência. A informação de frequência é alimentada em um modelo psicoacústico para determinar as frequências mascaradas. Então o orçamento de bits disponível é dividido entre as bandas, com mais bits alocados para o bandas com o poder espectral mais desmascarado, menos bits alocados para desmascarar bandas com menos potência espectral e sem bits alocados para bandas mascaradas. Finalmente, os bits são codificados usando a codificação Huffman, que atribui códigos curtos a numbers que aparecem com freqüência e códigos longos para aqueles que ocorrem com pouca freqüência. Lá

são muitos mais detalhes para o leitor curioso. Para obter mais informações, consulte Brandenburg (1999).

7.4.2 Vídeo Digital

Agora que sabemos tudo sobre o ouvido, é hora de passar ao olho. (Não este seção não é seguida por uma no nariz.) O olho humano tem a propriedade de quando uma imagem aparece na retina, a imagem é retida por um certo número de milissegundos antes de decair. Se uma sequência de imagens for desenhada a 50 imagens / s, o olho não percebe que está olhando para imagens discretas. Todos os sistemas de vídeo explorar este princípio para produzir imagens em movimento.

A representação digital mais simples de vídeo é uma sequência de quadros, cada formação de uma grade retangular de elementos de imagem, ou **pixels**. Cada pixel pode ser um bit único, para representar preto ou branco. No entanto, a qualidade de tal sistema tem é horrível. Tente usar seu editor de imagens favorito para converter os pixels de uma cor imagem para preto e branco (e *não* tons de cinza).

A próxima etapa é usar 8 bits por pixel para representar 256 níveis de cinza. Isto esquema fornece vídeo em "preto e branco" de alta qualidade. Para vídeo colorido, muitos sistemas usam 8 bits para cada uma das composições de cores primárias vermelho, verde e azul (RGB). Esta representação é possível porque qualquer cor pode ser construída a partir de um sobreposição linear de vermelho, verde e azul com as intensidades apropriadas. Com

24 bits por pixel, existem cerca de 16 milhões de cores, o que é mais do que o humano olho pode distinguir.

Em monitores e televisores LCD coloridos de computador, cada pixel discreto é feito de subpixels vermelhos, verdes e azuis próximos. Os quadros são exibidos por conjunto-ajustar a intensidade dos subpixels, e o olho mescla os componentes de cor.

As taxas de quadros comuns são de 24 quadros / s (herdadas de um filme de 35 mm filme), 30 quadros / s (herdados das televisões NTSC dos EUA) e 30 quadros / s (herdado do sistema de televisão PAL usado em quase todo o resto do mundo).

(Para os realmente exigentes, a televisão em cores NTSC funciona a 29,97 quadros / s. sistema preto e branco funcionava a 30 quadros / seg, mas quando a cor foi introduzida, o os engenheiros precisaram de um pouco de largura de banda extra para sinalização, então reduziram o quadro taxa para 29,97. Vídeos NTSC destinados a computadores realmente usam 30.) PAL foi em ventilado após NTSC e realmente usa 25.000 frames / seg. Para tornar esta história complete, um terceiro sistema, SECAM, é usado na França, África francófona e leste Europa. Foi introduzido pela primeira vez na Europa Oriental pela então comunista Alemanha Oriental.

muitos, de modo que o povo da Alemanha Oriental não podia assistir à televisão da Alemanha Ocidental (PAL) para que não tenham ideias ruins. Mas muitos desses países estão mudando para PAL. Tech-tecnologia e política no seu melhor.

Na verdade, para transmissão de televisão, 25 quadros / s não é bom o suficiente para movimento suave para que as imagens sejam divididas em dois **campos**, um com o número ímpar linhas de varredura e uma com as linhas de varredura pares. Os dois (meia resolução) campos são transmitidos sequencialmente, dando quase 60 (NTSC) ou exatamente 50 (PAL) campos / s, um sistema conhecido como **entrelaçamento**. Vídeos destinados à visualização em um computador são **progressivos**, isto é, não use entrelaçamento porque o computador monitores têm buffers em suas placas gráficas, tornando possível para a CPU colocar uma nova imagem no buffer 30 vezes / seg, mas a placa gráfica redesenha a tela 50 ou até 100 vezes / seg para eliminar a cintilação. Os aparelhos de televisão analógica não têm um buffer de quadros da maneira que os computadores fazem. Quando um vídeo entrelaçado com movimento rápido

mento é exibido em um computador, linhas horizontais curtas serão visíveis quase nítidas bordas, efeito conhecido como **pentejar**.

Os tamanhos de quadro usados para vídeo enviado pela Internet variam amplamente para o sim razão pela qual frames maiores requerem mais largura de banda, o que nem sempre pode ser acessível. O vídeo de baixa resolução pode ter 320 por 240 pixels e "tela inteira" o vídeo tem 640 por 480 pixels. Essas dimensões se aproximam daquelas dos primeiros monitores de computador e televisão NTSC, respectivamente. A **proporção** ou largura para proporção de altura, de 4: 3, é a mesma que uma televisão padrão. **HDTV** (alta definição) Os vídeos da TeleVision podem ser baixados com 1280 por 720 pixels. Estes As imagens "widescreen" têm uma proporção de 16: 9 para corresponder melhor a 3: 2 relação de aspecto do filme. Para comparação, o vídeo de DVD padrão é geralmente 720 por 480 pixels, e o vídeo em discos Blu-ray geralmente é HDTV de 1080 por 720 pixels.

Na Internet, o número de pixels é apenas parte da história, como a mídia jogadores podem apresentar a mesma imagem em tamanhos diferentes. O vídeo é apenas outra janela em uma tela de computador que pode ser aumentada ou reduzida. O papel de mais

Página 730

706

A CAMADA DE APLICAÇÃO INDIVÍDUO. 7

pixels é para aumentar a qualidade da imagem, para que não pareça borrada quando é expandido. No entanto, muitos monitores podem mostrar imagens (e, portanto, vídeos) com ainda mais pixels do que mesmo HDTV.

Compressão de Vídeo

Deve ser óbvio em nossa discussão sobre vídeo digital que a compressão é crítica para o envio de vídeo pela Internet. Até mesmo um vídeo de qualidade padrão com 640 por 480 pixels de quadros, 24 bits de informação de cor por pixel e 30 quadros / s ocupa mais de 200 Mbps. Isso excede em muito a largura de banda pela qual a maioria das empresas de

fices estão conectados à Internet, quanto mais usuários domésticos, e isso é para um único stream de vídeo. Já que a transmissão de vídeo descompactado está completamente fora do questão, pelo menos em redes de longa distância, a única esperança é que a compressão massiva é possível. Felizmente, um grande corpo de pesquisas nas últimas décadas

levou a muitas técnicas de compressão e algoritmos que fazem a transmissão de vídeo solução viável.

Muitos formatos são usados para o vídeo que é enviado pela Internet, alguns próprios eletário e algum padrão. A codificação mais popular é MPEG em seus vários formulários. É um padrão aberto encontrado em arquivos com extensões mpg e mp4, também como em outros formatos de contêiner. Nesta seção, veremos o MPEG para estudar como a compressão de vídeo é realizada. Para começar, veremos a compressão de imagens estáticas com JPEG. Um vídeo é apenas uma sequência de imagens (mais som).¹ Uma forma de compactar o vídeo é codificar cada imagem em sucessão. Para um primeiro aproximação, MPEG é apenas a codificação JPEG de cada quadro, além de alguns recursos extras para remover a redundância entre frames.

O padrão JPEG

O padrão **JPEG** (**Joint Photographic Experts Group**) para compactação de imagens estáticas de tom contínuo (por exemplo, fotografias) foram reveladas por fotográfico especialistas trabalhando sob os auspícios conjuntos da ITU, ISO e IEC, outras normas corporativas. É amplamente utilizado (procure arquivos com a extensão jpg) e frequentemente fornece taxas de compressão de 10: 1 ou melhores para imagens naturais.

JPEG é definido no International Standard 10918. Na verdade, é mais como uma lista de compras do que um único algoritmo, mas dos quatro modos que são definidos apenas o modo sequencial com perdas é relevante para nossa discussão. Além disso, vamos concentrar na forma como o JPEG é normalmente usado para codificar imagens de vídeo RGB de 24 bits

e deixará de fora algumas das opções e detalhes por uma questão de simplicidade.

O algoritmo é ilustrado na Figura 7.44. A etapa 1 é a preparação do bloco. Para o por questões de especificidade, vamos supor que a entrada JPEG é uma imagem RGB de 640×480 com 24 bits / pixel, conforme mostrado na Figura 7.44 (a). RGB não é o melhor modelo de cor para usar para compressão. O olho é muito mais sensível à **luminância**, ou seja, de sinais de vídeo do que a **crominância**, ou cor, de sinais de vídeo. Assim, nós

Página 731

SEC. 7,4

TRANSMISSÃO DE ÁUDIO E VÍDEO

707

primeiro calcule a luminância, Y , e as duas crominâncias, Cb e Cr , a partir de R , G e B . As fórmulas a seguir são usadas para valores de 8 bits que variam de 0 a 255:

$$Y = 16 + 0,26R + 0,50G + 0,09B$$

$$Cb = 128 + 0,15R - 0,29G - 0,44B$$

$$Cr = 128 + 0,44R - 0,37G + 0,07B$$

Quadra
preparação
Discreto
coseno
transformar
Quantização
Diferencial
quantização
Corre-
comprimento
codificação
Estatístico
resultado
codificação
Entrada
Resultado

Figura 7-44. Etapas na codificação sequencial com perdas de JPEG.

Matrizes separadas são construídas para Y , Cb e Cr . Em seguida, blocos quadrados de quatro pixels são calculados em média nas matrizes Cb e Cr para reduzi-los a 320×240 . Esta redução é com perdas, mas o olho mal percebe, pois o olho responde a luminância mais do que crominância. No entanto, ele comprime o total quantidade de dados por um fator de dois. Agora 128 é subtraído de cada elemento de todas as três matrizes para colocar 0 no meio do intervalo. Finalmente, cada matriz é

dividido em blocos de 8×8 . A matriz Y tem 4800 blocos; os outros dois têm 1200 blocos cada, conforme mostrado na Figura 7.45 (b).

```
480
640
(uma)
(b)
Cr
RGB
Y
Cb
640
480
240
320
240
1 bloco
Bloco 4799
Pixel de 8 bits
Pixel de 24 bits
```

Figura 7-45. (a) Dados de entrada RGB. (b) Após a preparação do bloco.

A etapa 2 da codificação JPEG é aplicar um **DCT** (**Discrete Cosine Transforma-ção**) para cada um dos 7200 blocos separadamente. A saída de cada DCT é um 8×8 matriz de coeficientes DCT. O elemento DCT (0, 0) é o valor médio do bloco.

Os outros elementos contam quanta potência espectral está presente em cada freqüência espacial freqüência. Normalmente, esses elementos decaem rapidamente com a distância da origem, (0, 0), conforme sugerido pela Fig. 7-46.

Uma vez que o DCT é concluído, a codificação JPEG segue para a etapa 3, chamada **quantização**, na qual os coeficientes DCT menos importantes são eliminados. Isso (com perdas)

Página 732

708

A CAMADA DE APLICAÇÃO
INDIVÍDUO. 7

```
Y / Cb / Cr
Amplitude
DCT
x
Fx
y
Fy
(uma)
(b)
```

Figura 7-46. (a) Um bloco da matriz Y . (b) Os coeficientes DCT.

a transformação é feita dividindo cada um dos coeficientes no DCT 8×8 matriz por um peso retirado de uma tabela. Se todos os pesos forem 1, a transformação faz nada. No entanto, se os pesos aumentam acentuadamente a partir da origem, maior as frequências espaciais são reduzidas rapidamente.

Um exemplo dessa etapa é dado na Figura 7.47. Aqui vemos o DCT inicial matriz, a tabela de quantização e o resultado obtido pela divisão de cada elemento DCT pelo elemento correspondente da tabela de quantização. Os valores no quantiza-ção não faz parte do padrão JPEG. Cada aplicativo deve fornecer seu próprio, permitindo-lhe controlar a compensação perda-compressão.

```
150
92
52
12
4
2
1
0
80
75
38
8
3
2
1
0
40
36
26
6
2
1
0
0
```


Figura 7-47. Cálculo dos coeficientes DCT quantizados.

A etapa 4 reduz o valor $(0, 0)$ de cada bloco (aquele no canto superior esquerdo) substituindo-o pela quantidade que difere do elemento correspondente no bloco anterior. Uma vez que esses elementos são as médias de seus respectivos blocos, eles devem mudar lentamente, então tomar os valores diferenciais deve reduzir a maior parte com valores pequenos. Nenhum diferencial é calculado a partir dos outros valores.

nique conhecido como **codificação de comprimento de execução**.

Figura 7-48. A ordem em que os valores quantizados são transmitidos.

Agora temos uma lista de números que representam a imagem (no espaço de transformação).

Etapa 6 Huffman-codifica os números para armazenamento ou transmissão, atribuindo com Mon números códigos mais curtos do que os incomuns.

JPEG pode parecer complicado, mas é porque é complicado. Ainda assim, os benefícios da compressão de até 20: 1 valem a pena. A decodificação de uma imagem JPEG requer

executando o algor

contanto que a codificação. Esta propriedade não é verdadeira para todo

compressão, pois nós
agora deve ver.

os padrões definem os principais algoritmos usados para compactar vídeos. Eles têm sido padrões internacionais desde 1993. Porque os filmes contêm imagens e som, o MPEG pode comprimir áudio e vídeo. Já examinamos compressão de áudio e compressão de imagem estática, então vamos agora examinar o vídeo compressão. O padrão MPEG-1 (que inclui áudio MP3) foi publicado pela primeira vez em 1993 e ainda é amplamente utilizado. Seu objetivo era produzir gravador de vídeo com qualidade de saída colocar que foi comprimido 40: 1 para taxas de cerca de 1 Mbps. Este vídeo é adequado para

Página 734

710

A CAMADA DE APLICAÇÃO INDIVÍDUO. 7

amplo uso da Internet em sites. Não se preocupe se você não se lembra do vídeo gravadores - MPEG-1 também era usado para armazenar filmes em CDs quando existiam. Se você não sabe o que são CDs, teremos que passar para o MPEG-2. O padrão MPEG-2, lançado em 1996, foi projetado para compactar vídeo com qualidade de transmissão. É muito comum agora, pois é usado como base para vídeos codificados em DVDs (que inevitavelmente encontram seu caminho na Internet) e para digital transmissão de televisão (como DVB). Vídeo com qualidade de DVD normalmente é codificado em taxas de 4–8 Mbps.

O padrão MPEG-4 possui dois formatos de vídeo. O primeiro formato, lançado em 1999, codifica o vídeo com uma representação baseada em objeto. Isso permite a mistura de imagens naturais e sintéticas e outros tipos de mídia, por exemplo, um meteorologista em frente a um mapa meteorológico. Com esta estrutura, é fácil deixe os programas interagirem com os dados do filme. O segundo formato, lançado em 2003, é conhecido como **H.264** ou **AVC (Advanced Video Coding)**. Seu objetivo é codificar vídeo pela metade da taxa de codificadores anteriores para o mesmo nível de qualidade, para melhor fornecer porta a transmissão de vídeo em redes. Este codificador é usado para HDTV em a maioria dos discos Blu-ray.

Os detalhes de todos esses padrões são muitos e variados. Os padrões posteriores também tem muito mais recursos e opções de codificação do que os padrões anteriores. No entanto, não entraremos em detalhes. Na maior parte, os ganhos em vídeo compressão ao longo do tempo veio de inúmeras pequenas melhorias, ao invés do que mudanças fundamentais em como o vídeo é compactado. Assim, vamos esboçar o conceitos gerais.

MPEG compacta áudio e vídeo. Já os codificadores de áudio e vídeo trabalhar de forma independente, há um problema de como os dois fluxos são sincronizados em o receptor. A solução é ter um único relógio que emite carimbos de data / hora do hora atual para ambos os codificadores. Esses carimbos de data / hora estão incluídos na saída codificada

colocados e propagados até o receptor, que pode usá-los para sincronizar os fluxos de áudio e vídeo.

A compressão de vídeo MPEG tira proveito de dois tipos de redundâncias que existem em filmes: espacial e temporal. A redundância espacial pode ser utilizada por sim- ply codificando cada quadro separadamente com JPEG. Esta abordagem é usada ocasionalmente, especialmente quando o acesso aleatório a cada quadro é necessário, como na edição de produtos de vídeo

ações. Neste modo, os níveis de compressão JPEG são alcançados.

A compressão adicional pode ser alcançada aproveitando o fato de que os quadros consecutivos costumam ser quase idênticos. Este efeito é menor do que poderia aparecer pela primeira vez, já que muitos diretores de cinema cortam as cenas a cada 3 ou 4 segundos (cronometrar um fragmento de filme e contar as cenas). No entanto, corridas de 75 ou mais

quadros altamente semelhantes oferecem o potencial de uma grande redução em vez de simplesmente encodificação de cada quadro separadamente com JPEG.

Para cenas em que a câmera e o fundo estão parados e um ou dois atores estão se movendo lentamente, quase todos os pixels serão idênticos no quadro enquadrar. Aqui, apenas subtraindo cada quadro do anterior e executando

Página 735

SEC. 7,4
TRANSMISSÃO DE ÁUDIO E VÍDEO

711

JPEG na diferença faria bem. No entanto, para cenas onde a câmera está panorâmica ou zoom, essa técnica falha seriamente. O que é necessário é alguma maneira de compensar este movimento. Isso é exatamente o que o MPEG faz; é a principal diferença entre MPEG e JPEG.

A saída MPEG consiste em três tipos de quadros:

1. Quadros I- (intracodificados): imagens estáticas compactadas autocontidas.

2. Quadros P- (preditivos): diferença bloco a bloco com o anterior quadros.

3. Quadros B- (bidirecionais): diferenças bloco a bloco entre os anteriores ou quadros futuros.

I-frames são apenas imagens estáticas. Eles podem ser codificados com JPEG ou algo assim semelhante. É importante ter I-frames aparecendo no fluxo de saída periodicamente (por exemplo, uma ou duas vezes por segundo) por três razões. Primeiro, o MPEG pode ser usado para uma transmissão multicast, com os telespectadores sintonizando à vontade. Se todos os quadros dependessem de seus predecessores voltando ao primeiro quadro, qualquer um que perdeu o primeiro frame nunca poderia decodificar quaisquer frames subsequentes. Em segundo lugar, se algum quadro fosse erroneamente, nenhuma decodificação adicional seria possível: tudo a partir de então seria lixo ininteligível. Terceiro, sem I-frames, ao fazer um avanço rápido ou retroceder o decodificador teria que calcular cada quadro passado para que ele soubesse o valor total daquele em que parou.

Os quadros P, em contraste, codificam as diferenças entre os quadros. Eles são baseados na ideia de **macroBlocos**, que cobrem, por exemplo, 16×16 pixels no espaço de luminância e 8×8 pixels no espaço de crominância. Um macrobloco é codificado pesquisando o quadro anterior para ele ou algo apenas ligeiramente diferente dele.

Um exemplo de onde os quadros P seriam úteis é dado na Figura 7.49. Aqui vemos três frames consecutivos que têm o mesmo fundo, mas diferem na posição de uma pessoa. Os macroBlocos contendo a cena de fundo irão correspondem exatamente, mas os macroBlocos contendo a pessoa serão compensados na posição por uma quantia desconhecida e terá que ser rastreada.

Figura 7-49. Três quadros consecutivos.

Os padrões MPEG não especificam como pesquisar, até que ponto pesquisar ou como uma boa combinação tem que ser para contar. Isso depende de cada implementação. Para

Página 736

712

A CAMADA DE APLICAÇÃO
INDIVÍDUO. 7

exemplo, uma implementação pode procurar um macrobloco na posição atual no quadro anterior, e todas as outras posições se deslocam $\pm \Delta x$ na direção x e $\pm \Delta y$ na direção y . Para cada posição, o número de correspondências na luminância matriz pode ser calculada. A posição com a maior pontuação seria declarada o vencedor, desde que esteja acima de algum limite predefinido. Caso contrário, o

seria dito que o macrobloco estava faltando. Algoritmos muito mais sofisticados são também possível, é claro.

Se um macrobloco for encontrado, ele é codificado pela diferença entre seus valor atual e o do quadro anterior (para luminância e ambos finanças). Essas matrizes de diferença são, então, submetidas ao coseno discreto transformação, quantização, codificação run-length e codificação Huffman, como de costume. O valor para o macrobloco no fluxo de saída é então o movimento vector (o quanto o macrobloco se moveu de sua posição anterior em cada direção), seguido pela codificação de sua diferença. Se o macrobloco não estiver localizado no quadro anterior, o valor atual é codificado, assim como em um quadro I.

Claramente, esse algoritmo é altamente assimétrico. Uma implementação é gratuita para testar cada posição plausível no quadro anterior se quiser, em uma tentativa desesperada para localizar cada macrobloco, não importa para onde ele foi movido. Esta abordagem irá minimizar o fluxo MPEG codificado às custas de uma codificação muito lenta.

Esta abordagem pode ser adequada para uma codificação única de uma biblioteca de filmes, mas ser terrível para videoconferências em tempo real.

Da mesma forma, cada implementação é livre para decidir o que constitui um "encontrado" macrobloco. Essa liberdade permite que os implementadores concorram na qualidade e velocidade de seus algoritmos, mas sempre produzem saída MPEG compatível.

Até agora, a decodificação de MPEG é simples. A decodificação de quadros I é semelhante a decodificação de imagens JPEG. A decodificação de P-frames requer que o decodificador armazene

o

quadros anteriores para que ele possa construir o novo em um buffer separado baseado em macroblocos codificados e macroblocos contendo diferenças dos anteriores quadros. O novo quadro é montado macrobloco por macrobloco.

Os quadros B são semelhantes aos quadros P, exceto que permitem a macro de referência bloco para estar nos quadros anteriores ou posteriores. Este adicional grátil dom permite uma melhor compensação de movimento. É útil, por exemplo, quando objetos passam na frente ou atrás de outros objetos. Para fazer a codificação B-frame, o en- o codificador precisa manter uma sequência de quadros na memória de uma vez: quadros anteriores, o

alugar o quadro sendo codificado e os quadros futuros. A decodificação é similarmente mais complicated e adiciona algum atraso. Isso ocorre porque um determinado quadro B não pode ser decodificado

até que os quadros sucessivos dos quais depende sejam decodificados. Assim, embora B- frames dão a melhor compressão, nem sempre são usados devido ao seu maior complexidade e requisitos de buffer.

Os padrões MPEG contêm muitos aprimoramentos para essas técnicas para atingir níveis excelentes de compressão. AVC pode ser usado para compactar vídeo em proporções superiores a 50: 1, o que reduz os requisitos de largura de banda da rede pelo mesmo fator. Para obter mais informações sobre AVC, consulte Sullivan e Wiegand (2005).

7.4.3 Streaming de mídia armazenada

Vamos agora passar para os aplicativos de rede. Nossa primeiro caso é o streaming dia que já está armazenado em arquivos. O exemplo mais comum disso é assistir vídeos na Internet. Esta é uma forma de **VoD** (Video on Demand). De outras formas de vídeo sob demanda usam uma rede de provedor que é separada do Inter- rede para entregar os filmes (por exemplo, a rede a cabo).

Na próxima seção, veremos streaming de mídia ao vivo, por exemplo, transmitir rádio IPTV e Internet. Então, vamos olhar para o terceiro caso de con em tempo real ferenciamento. Um exemplo é uma chamada de voz sobre IP ou videoconferência com Skype. Esses três casos colocam requisitos cada vez mais rigorosos sobre como podemos entregar er o áudio e o vídeo na rede porque devemos prestar atenção cada vez mais

para atrasar e tremer.

A Internet está cheia de sites de música e vídeo que transmitem arquivos de mídia armazenados. Na verdade, a maneira mais fácil de lidar com a mídia armazenada *não* é transmiti-la. Imagina você deseja criar um site de aluguel de filmes online para competir com o iTunes da Apple. A regolar permitirá que os usuários baixem e assistam a vídeos (depois de pagarem, de curso). A sequência de etapas é mostrada na Figura 7.50. Vamos soletrá-los para compare-los com o próximo exemplo.

3: Salvar
meios de comunicação
2: Resposta da mídia (HTTP)
1: Solicitação de mídia (HTTP)
Navegador
Cliente
meios de comunicação
jogador
Rede
servidor
Servidor
4: Jogar
meios de comunicação
Disco
Disco

Figura 7-50. Reprodução de mídia na Web por meio de downloads simples.

O navegador entra em ação quando o usuário clica em um filme. Na etapa 1, envia uma solicitação HTTP do filme para o servidor Web para o qual o filme está vinculado. Na etapa 2, o servidor busca o filme (que é apenas um arquivo MP4 ou algum outro formato) e o envia de volta ao navegador. Usando o tipo MIME, por exemplo, *video / mp4*, o navegador verifica como deve exibir o arquivo. No neste caso, é com um reproduutor de mídia que é mostrado como um aplicativo auxiliar, embora também pode ser um plug-in. O navegador salva o filme inteiro em um arquivo de trabalho em disco na etapa 3. Em seguida, inicia o reproduutor de mídia, passando-lhe o nome do scratch Arquivo. Finalmente, na etapa 4, o reproduutor de mídia começa a ler o arquivo e a reproduzir o filme.

Em princípio, essa abordagem é totalmente correta. Vai passar o filme. Lá não é um problema de rede em tempo real para resolver, porque o download é simplesmente um

Página 738

714

A CAMADA DE APLICAÇÃO INDIVÍDUO. 7

Download do arquivo. O único problema é que todo o vídeo deve ser transmitido por a rede antes do filme começar. A maioria dos clientes não quer esperar uma hora para seu "vídeo sob demanda". Este modelo pode ser problemático até mesmo para áudio. Eu estou agindo visualizando uma música antes de comprar um álbum. Se a música tiver 4 MB, que é um tamanho típico para uma música MP3, e a conectividade de banda larga é de 1 Mbps, o usuário será saudado por meio minuto de silêncio antes do início da visualização. Isto modelo provavelmente não venderá muitos álbuns.

Para contornar este problema sem alterar o funcionamento do navegador, os sites pode usar o design mostrado na Fig. 7-51. A página vinculada ao filme não é o arquivo de filme real. Em vez disso, é o que é chamado de **meta**-arquivo, um arquivo muito curto apenas na-

ming o filme (e possivelmente tendo outros descritores chave). Um meta-arquivo simples pode ser apenas uma linha de texto ASCII e ter a seguinte aparência:

`rtsp://joes-movie-server/movie-0025.mp4`

O navegador obtém a página normalmente, agora um arquivo de uma linha, nas etapas 1 e 2. Em seguida,

inicia o reproduutor de mídia e entrega o arquivo de uma linha na etapa 3, como de costume. o media player lê o metarquivo e vê a URL de onde obter o filme. isto entra em contato com *joes-video-server* e pede o filme na etapa 4. O filme é então transmitido de volta para o reproduutor de mídia na etapa 5. A vantagem deste arranjo é que o media player é iniciado rapidamente, após apenas um pequeno metarquivo ser baixado-ed. Quando isso acontece, o navegador não está mais no loop. A mídia é enviada

diretamente para o media player, que pode começar a exibir o filme antes de todo arquivo foi baixado.

- 4: Solicitação de mídia (RTSP)
 - 2: Resposta de metarquivo (HTTP)
 - 1: solicitação de metarquivo (HTTP)
- Navegador
Cliente
meios de comunicação
jogador
Rede
servidor
Servidor
meios de comunicação
servidor
Servidor
3: Handoff
metarquivo
5: Resposta da mídia (via TCP ou UDP)

Disco

Figura 7-51. Streaming de mídia usando a Web e um servidor de mídia.

Mostramos dois servidores na Figura 7.51 porque o servidor nomeado no metarquivo geralmente não é o mesmo que o servidor web. Na verdade, geralmente não é mesmo

Página 739

SEC. 7,4

TRANSMISSÃO DE ÁUDIO E VÍDEO

715

um servidor HTTP, mas um servidor de mídia especializado. Neste exemplo, o servidor de mídia usa **RTSP** (**Real Time Streaming Protocol**), conforme indicado pelo nome do esquema *rtsp*.

O media player tem quatro funções principais a fazer:

1. Gerenciar a interface do usuário.
2. Trate os erros de transmissão.
3. Descompacte o conteúdo.
4. Elimine o jitter.

A maioria dos reprodutores de mídia hoje em dia tem uma interface de usuário chamativa, às vezes simulando

uma unidade estéreo, com botões, botões, controles deslizantes e telas visuais. Freqüentemente há painéis frontais intercambiáveis, chamados de **skins**, que o usuário pode colocar no player.

O media player deve gerenciar tudo isso e interagir com o usuário.

Os outros trabalhos estão relacionados e dependem dos protocolos de rede. Nós iremos através de cada um, começando com o tratamento de erros de transmissão. Lidando com erros dependem de se um transporte baseado em TCP como HTTP é usado para transportar a mídia ou um transporte baseado em UDP como RTP é usado. Ambos são usados na prática.

Se um transporte baseado em TCP estiver sendo usado, não haverá erros para a mídia jogador para corrigir porque o TCP já fornece confiabilidade usando retransmissão sessões. Esta é uma maneira fácil de lidar com erros, pelo menos para o media player, mas complica a remoção do jitter em uma etapa posterior.

Alternativamente, um transporte baseado em UDP como RTP pode ser usado para mover os dados. Nós o estudamos no cap. 6. Com esses protocolos, não há retransmissões.

Assim, a perda de pacotes devido a congestionamento ou erros de transmissão significará que alguns dos

a mídia não chega. Cabe ao media player lidar com esse problema.

Vamos entender a dificuldade que enfrentamos. A perda é um problema porque porque os clientes não gostam de grandes lacunas em suas músicas ou filmes. No entanto, é não é um problema tanto quanto a perda em uma transferência regular de arquivos, porque a perda de um

pequena quantidade de mídia não precisa degradar a apresentação para o usuário. Para vídeo, o usuário provavelmente não perceberá se ocasionalmente houver 24 novos quadros em algum segundo

ond em vez de 25 novos quadros. Para áudio, pequenos intervalos no playout podem ser mascarados com sons próximos no tempo. É improvável que o usuário detecte essa substituição, a menos eles estão prestando *muita* atenção.

A chave para o raciocínio acima, entretanto, é que as lacunas são muito curtas. Internet-congestionamento de trabalho ou um erro de transmissão geralmente fará com que um pacote inteiro seja perdido, e os pacotes são freqüentemente perdidos em pequenas explosões. Duas estratégias podem ser usadas para reduzir o impacto da perda de pacotes na mídia perdida: FEC e intercalação. Nós irá descrever cada um por vez.

FEC (**Forward Error Correction**) é simplesmente a codificação de correção de erros que nós estudamos no cap. 3 aplicado no nível do aplicativo. Paridade entre pacotes provides um exemplo (Shacham e McKenny, 1990). Para cada quatro pacotes de dados

Página 740

716

A CAMADA DE APLICAÇÃO INDIVÍDUO. 7

que são enviados, um quinto **pacote de paridade** pode ser construído e enviado. Isso é mostrado em Fig. 7-52 com os pacotes de A , B , C , e D . O pacote de paridade, P , contém redundantes bits que são as somas de paridade ou OU exclusivo dos bits em cada um dos quatro dados pacotes. Esperançosamente, todos os pacotes chegarão para a maioria dos grupos de cinco pacotes. Quando isso acontece, o pacote de paridade é simplesmente descartado no receptor. Ou se apenas o pacote de paridade é perdido, nenhum dano é causado.

Pacote perdido
Construir paridade:
Cliente
meios de comunicação
jogador
meios de comunicação
servidor
Servidor
Perda de reparo:
Pacote de paridade
= P
UMA
B
+
+
C D +
UMA
C
D
P
B
= A
B
P
+
+
C D +
Disco

Figura 7-52. Usando um pacote de paridade para reparar a perda.

Ocasionalmente, no entanto, um pacote de dados pode ser perdido durante a transmissão, pois B é na Fig. 7-52. O media player recebe apenas três pacotes de dados, A , C e D , mais o pacote de paridade, P . Por design, os bits no pacote de dados ausente podem ser reconstruído a partir dos bits de paridade. Para ser mais específico, usando " + " para representar OU exclusivo

ou adição do módulo 2, B pode ser reconstruído como $B = P + A + C + D$ pelo próprio laços de OU exclusivo (ou seja, $X + Y + Y = X$).

O FEC pode reduzir o nível de perda visto pelo media player reparando alguns das perdas de pacotes, mas só funciona até um certo nível. Se dois pacotes em um grupo de cinco está perdido, não há nada que possamos fazer para recuperar os dados. O outro a propriedade a ser observada sobre o FEC é o custo que pagamos para obter essa proteção.

Cada quatro pacotes se transformam em cinco pacotes, portanto, os requisitos de largura de banda para

a mídia é 25% maior. A latência de decodificação também aumentou, como podemos precisamos esperar até que o pacote de paridade chegue antes de podermos reconstruir um dado pacote que veio antes dele.

Há também um truque inteligente na técnica acima. No cap. 3, nós de-paridade riscada como fornecendo detecção de erro. Aqui estamos fornecendo error-correção. Como pode fazer as duas coisas? A resposta é que, neste caso, sabe-se qual pacote foi perdido. Os dados perdidos são chamados de **apagamento**. No cap. 3, quando consideramos

eramos um frame que foi recebido com alguns bits errados, não sabíamos qual bit estava errado. Este caso é mais difícil de lidar do que rasuras. Assim, com rasuras a paridade pode fornecer correção de erros e, sem apagamentos, a paridade só pode fornecer detecção de erro. Veremos outro benefício inesperado da paridade em breve, quando chegar a cenários multicast.

A segunda estratégia é chamada de **intercalação**. Esta abordagem é baseada na mistura acima ou intercalando a ordem da mídia antes da transmissão e descompactação ou

Página 741

SEC. 7,4
TRANSMISSÃO DE ÁUDIO E VÍDEO

717

desintercalando-o na recepção. Dessa forma, se um pacote (ou rajada de pacotes) for perdido, a perda será distribuída ao longo do tempo pela remoção da mistura. Não resultará em um único, grande lacuna quando a mídia é reproduzida. Por exemplo, um pacote pode conter 220 amostras estéreo, cada uma contendo um par de números de 16 bits, normalmente bom para 5 msec de música. Se as amostras foram enviadas em ordem, um pacote perdido representaria um 5 ms de intervalo na música. Em vez disso, as amostras são transmitidas conforme mostrado na Fig. 7-

53. Todas as amostras pares para um intervalo de 10 ms são enviadas em um pacote, seguido por todas as amostras ímpares no próximo. A perda do pacote 3 agora não representa ent um intervalo de 5 ms na música, mas a perda de todas as outras amostras por 10 ms. Esta perda pode ser tratada facilmente fazendo com que o media player interpole usando o amostras anteriores e posteriores. O resultado é uma resolução temporal mais baixa para 10 msec, mas não um intervalo de tempo perceptível na mídia.

Mesmo tempo

amostra

lenda

Tempo ímpar

amostra

0

(b)

Pacote

(uma)

5

0

10

15

Tempo (msec)

20

Perdido

Este pacote contém

220 amostras de tempo par

25

30

1

2

4

5

Este pacote contém

220 amostras de tempo ímpar

Figura 7-53. Quando os pacotes carregam amostras alternativas, a perda de um pacote reduz a resolução temporal em vez de criar uma lacuna no tempo.

Este esquema de intercalação acima funciona apenas com amostragem não compactada.

No entanto, a intercalação (em curtos períodos de tempo, não em amostras individuais) pode também pode ser aplicado após a compressão, desde que haja uma maneira de encontrar o limite da amostra

áries no fluxo comprimido. RFC 3119 fornece um esquema que funciona com áudio compactado.

A intercalação é uma técnica atraente quando pode ser usada porque não precisa

largura de banda adicional, ao contrário do FEC. No entanto, a intercalação aumenta a latência, apenas como o FEC, devido à necessidade de esperar a chegada de um grupo de pacotes (para que possam ser desintercalados).

A terceira tarefa do media player é descompactar o conteúdo. Embora esta tarefa é computacionalmente intensivo, é bastante simples. A questão espinhosa é como para decodificar a mídia se o protocolo de rede não corrigir erros de transmissão. No muitos esquemas de compressão, os dados posteriores não podem ser descompactados até o anterior dados foram descompactados, porque os dados posteriores são codificados em relação ao ouvido dizer. Para um transporte baseado em UDP, pode haver perda de pacotes. Assim, a codificação

718

A CAMADA DE APLICAÇÃO INDIVÍDUO. 7

o processo deve ser projetado para permitir a decodificação apesar da perda de pacotes. Este requer-É por isso que o MPEG usa quadros I, P e B. Cada quadro I pode ser decodificado independentemente

pendentemente dos outros quadros para se recuperar da perda de quaisquer quadros anteriores. A quarta tarefa é eliminar o jitter, a maldição de todos os sistemas de tempo real. o solução geral que descrevemos na Seç. 6.4.3 é usar um buffer de playout. Todos sistemas de streaming começam armazenando em buffer 5-10 segundos de mídia antes de começar a jogar, como mostrado na Fig. 7-54. A reprodução drena a mídia regularmente do buffer, então se o áudio está claro e o vídeo é bom. O atraso de inicialização dá ao buffer uma chance de preencher até a **marca d'água baixa**. A ideia é que agora os dados cheguem regularmente o suficiente para que o buffer nunca seja completamente esvaziado. Se isso acontecesse caneta, a exibição de mídia iria parar. O valor do buffer é que, se os dados forem às vezes demora para chegar devido ao congestionamento, a mídia em buffer permitirá que a reprodução continuará normalmente até que uma nova mídia chegue e o buffer seja reabastecido.

Amortecedor

Baixo-

água

marca

Alto-

água

marca

meios de comunicação

jogador

meios de comunicação

servidor

Máquina cliente

Máquina servidor

Figura 7-54. O media player armazena em buffer a entrada do servidor de mídia e reproduz do buffer em vez de diretamente da rede.

Quanto buffer é necessário e a rapidez com que o servidor de mídia envia mídia para encher o buffer, depende dos protocolos de rede. Existem muitas possibilidades.

O maior fator no projeto é se um transporte baseado em UDP ou um baseado em TCP transporte é usado.

Suponha que um transporte baseado em UDP como RTP seja usado. Além disso, suponha que há ampla largura de banda para enviar pacotes do servidor de mídia para a mídia jogador com pouca perda e pouco outro tráfego na rede. Neste caso, os pacotes pode ser enviado na taxa exata em que a mídia está sendo reproduzida. Cada pacote irá trans-
sentar na rede e, após um atraso de propagação, chegar aproximadamente no momento certo para o media player para apresentar a mídia. É necessário muito pouco armazenamento em buffer, pois não há

variabilidade no atraso. Se intercalação ou FEC for usado, mais buffer é necessário para pelo menos o grupo de pacotes sobre os quais a intercalação ou FEC é executada. Como sempre, isso adiciona apenas uma pequena quantidade de buffer.

Infelizmente, esse cenário não é realista em dois aspectos. Primeiro, largura de banda varia de acordo com os caminhos da rede, por isso geralmente não fica claro para o servidor de mídia se

haverá largura de banda suficiente antes de tentar transmitir a mídia. Um simples solução é codificar mídia em múltiplas resoluções e deixar cada usuário escolher um

Página 743

SEC. 7,4

TRANSMISSÃO DE ÁUDIO E VÍDEO

719

resolução que é suportada por sua conectividade com a Internet. Freqüentemente, há apenas dois níveis: alta qualidade, digamos, codificado em 1,5 Mbps ou melhor, e baixa qualidade, digamos encodificado a 512 kbps ou menos.

Em segundo lugar, haverá algum jitter, ou variação em quanto tempo leva o samples para cruzar a rede. Esse jitter vem de duas fontes. Muitas vezes há uma quantidade apreciável de tráfego competitivo na rede - alguns dos quais podem vêm de usuários multitarefa navegando na Web enquanto ostensivamente assistir a um filme transmitido). Este tráfego causará flutuações quando a mídia chega. Além disso, nos preocupamos com a chegada de quadros de vídeo e áudio amostras, não pacotes. Com a compressão, os quadros de vídeo em particular podem ser maiores ou menor dependendo de seu conteúdo. Uma sequência de ação normalmente levará mais bits para codificar do que uma paisagem plácida. Se a largura de banda da rede for constante, a taxa de entrega de mídia em relação ao tempo irá variar. Quanto mais jitter, ou variação atraso, a partir dessas fontes, quanto maior a marca d'água baixa do buffer precisa ser para evitar underrun.

Agora, suponha que um transporte baseado em TCP como HTTP seja usado para enviar a mídia. Executando retransmissões e esperando para entregar os pacotes até que eles estejam em ordem, o TCP aumentará o jitter que é observado pelo media player, talvez significativamente. O resultado é que um buffer maior e uma marca d'água baixa mais alta são necessário. No entanto, existe uma vantagem. TCP enviará dados tão rápido quanto a rede vai carregá-lo. Às vezes, a mídia pode atrasar se a perda precisar ser reparada. Mas muito do tempo, a rede será capaz de entregar mídia mais rápido do que o jogador soma isso. Nestes períodos, o buffer irá encher e prevenir subexecuções futuras. Se a rede é significativamente mais rápida do que a taxa média de mídia, como costuma ser o caso, o buffer irá encher rapidamente após a inicialização, de modo que o esvaziamento logo deixará de ser um problema.

Com TCP ou UDP e uma taxa de transmissão que excede a taxa de playout, uma questão é o quanto longe do ponto de playout o reproduutor de mídia e o servidor de mídia estão dispostos a prosseguir. Freqüentemente, eles desejam fazer o download do arquivo inteiro. No entanto, avançar muito à frente do ponto de playout realiza um trabalho que não é ainda necessário, pode exigir armazenamento significativo e não é necessário para evitar buffer underruns. Quando não é desejado, a solução é o media player definir uma marca d'água alta no buffer. Basicamente, o servidor apenas bombeia dados até que o buffer é preenchido até a marca d'água alta. Em seguida, o media player diz para pausar. Uma vez que os dados continuarão a chegar até que o servidor tenha obtido o pedido de pausa, a distância entre a marca d'água alta e o final do buffer deve ser maior do que o produto de atraso de largura de banda da rede. Depois que o servidor tiver parado, o buffer começará a esvaziar. Quando atinge a marca d'água baixa, o media player diz ao servidor de mídia para iniciar novamente. Para evitar underrun, o low-watermark ter também deve levar em consideração o produto de atraso de largura de banda da rede ao solicitar ao servidor de mídia para retomar o envio da mídia.

Para iniciar e interromper o fluxo de mídia, o media player precisa de um controle remoto para isso. Isso é o que o RTSP oferece. É definido no RFC 2326 e fornece o

Página 744

720

A CAMADA DE APLICAÇÃO
INDIVÍDUO. 7

mecanismo para o jogador controlar o servidor. Bem como iniciar e parar o fluxo, ele pode voltar ou avançar para uma posição, reproduzir intervalos especificados e jogar em velocidades rápidas ou lentas. Ele não fornece para o fluxo de dados, porém, que geralmente é RTP sobre UDP ou RTP sobre HTTP sobre TCP.

Os principais comandos fornecidos pelo RTSP estão listados na Figura 7.55. Eles TEM um formato de texto simples, como mensagens HTTP, e geralmente são transportados por TCP. RTSP pode ser executado sobre UDP também, uma vez que cada comando é reconhecido (e assim pode ser reenviado se não for confirmado).

Comando

Ação do servidor

DESCREVER

Parâmetros de lista de mídia

CONFIGURAÇÃO

Estabeleça um canal lógico entre o jogador e o servidor

TOQUE

Comece a enviar dados para o cliente

REGISTRO

Comece a aceitar dados do cliente

PAUSA

Pare temporariamente de enviar dados

DESTRUÍR

Libere o canal lógico

Figura 7-55. Comandos RTSP do jogador para o servidor.

Mesmo que o TCP pareça um ajuste inadequado para o tráfego em tempo real, ele é frequentemente usado em

prática. O principal motivo é que ele é capaz de passar por firewalls com mais facilidade do que UDP, especialmente quando executado na porta HTTP. A maioria dos administradores config-

ure firewalls para proteger suas redes de visitantes indesejados. Eles quase todos maneiras permitem que conexões TCP da porta remota 80 passem para HTTP e Tráfego da web. O bloqueio dessa porta leva rapidamente a campistas infelizes. No entanto, a maioria

outras portas estão bloqueadas, incluindo para RSTP e RTP, que usam as portas 554 e 5004, entre outros. Assim, a maneira mais fácil de obter mídia de streaming por meio do firewall é para o site fingir que é um servidor HTTP enviando um

Resposta HTTP, pelo menos para o firewall.

Existem algumas outras vantagens do TCP também. Porque fornece confiabilidade, O TCP fornece ao cliente uma cópia completa da mídia. Isso torna mais fácil para um usuário para retroceder até um ponto de playout visualizado anteriormente sem se preocupar com a perda de dados.

Por fim, o TCP armazenará em buffer o máximo de mídia possível, o mais rápido possível.

Quando o espaço do buffer é barato (o que acontece quando o disco é usado para armazenamento), o media player pode baixar a mídia enquanto o usuário assiste. Assim que o download estiver completo, o usuário poderá assistir sem interrupções, mesmo que perca a conectividade. Isto propriedade é útil para celulares porque a conectividade pode mudar rapidamente com movimento.

A desvantagem do TCP é a latência de inicialização adicionada (por causa do TCP inicialização) e também uma marca d'água mais alta. No entanto, isso raramente é muito penalidade, desde que a largura de banda da rede exceda a taxa de mídia por um grande fator.

7.4.4 Streaming Live Media

Não são apenas os vídeos gravados que são extremamente populares na web. Viver o streaming de mídia também é muito popular. Assim que se tornou possível transmitir áudio e vídeo na Internet, rádios comerciais e estações de TV tiveram a ideia de transmitir seu conteúdo pela Internet e também pelo ar. Não muito depois isso, as estações universitárias começaram a enviar seus sinais pela Internet. Então col-

os *alunos de lege* começaram suas próprias transmissões pela Internet.

Hoje, pessoas e empresas de todos os tamanhos transmitem áudio e vídeo ao vivo. o

A área é um foco de inovação à medida que as tecnologias e os padrões evoluem. Viver streaming é usado para uma presença online pelas principais estações de televisão. Esta é a chamada-
ed **IPTV (IP TeleVision)**. Ele também é usado para transmitir estações de rádio como a BBC.

Isso é chamado de **rádio na Internet** . Tanto a IPTV quanto o rádio na Internet alcançam o público em todo o mundo para eventos que vão desde desfiles de moda a futebol da Copa do Mundo e testes jogos ao vivo do Melbourne Cricket Ground. A transmissão ao vivo sobre IP é usada como uma tecnologia de provedores de cabo para construir seus próprios sistemas de transmissão. E isso é

amplamente utilizado por operações de baixo orçamento de sites adultos a zoológicos. Com a tecnologia atual

tecnologia, virtualmente qualquer pessoa pode iniciar a transmissão ao vivo rapidamente e com poucos gastos.

Uma abordagem para streaming ao vivo é gravar programas em disco. Os espectadores podem conecte-se aos arquivos do servidor, abra qualquer programa e baixe-o para ouvir ing. Um **podcast** é um episódio recuperado dessa maneira. Para eventos programados, é também é possível armazenar conteúdo logo após ser transmitido ao vivo, de modo que o arquivo é apenas

correndo, digamos, meia hora ou menos atrás do feed ao vivo.

Na verdade, essa abordagem é exatamente a mesma usada para streaming de mídia acabamos de discutir. É fácil de fazer, todas as técnicas que discutimos funcionam para e os visualizadores podem escolher entre todos os programas no arquivo.

Uma abordagem diferente é transmitir ao vivo pela Internet. Os espectadores sintonizam um fluxo contínuo de mídia, exatamente como ligar a televisão. No entanto, a mídia os jogadores fornecem os recursos adicionais de permitir que o usuário pause ou retroceda o playout. A mídia ao vivo continuará a ser transmitida e será armazenada em buffer pelo jogador até que o usuário esteja pronto para isso. Do ponto de vista do navegador, parece exatamente como o caso de streaming de mídia armazenada. Não importa para o jogador se o conteúdo vem de um arquivo ou está sendo enviado ao vivo, e geralmente o player não ser capaz de saber (exceto que não é possível pular para a frente com uma transmissão ao vivo). Dada a semelhança do mecanismo, muito da nossa discussão anterior se aplica, mas também existem algumas diferenças importantes.

É importante ressaltar que ainda há a necessidade de armazenamento em buffer no lado do cliente para suavizar

jitter. Na verdade, uma quantidade maior de buffer é frequentemente necessária para streaming ao vivo (

dependente da consideração de que o usuário pode pausar a reprodução). Quando transmitir-
de um arquivo, a mídia pode ser empurrada a uma taxa maior do que a
taxa de volta. Isso criará um buffer rapidamente para compensar o jitter da rede
(e o reprodutor interromperá a transmissão se não quiser armazenar mais dados em buffer). No
Em contraste, o streaming de mídia ao vivo é sempre transmitido com precisão na taxa que está

722

A CAMADA DE APLICAÇÃO INDIVÍDUO. 7

gerado, que é igual à taxa de reprodução. Não pode ser
enviado mais rápido do que isso. Como consequência, o buffer deve ser grande o suficiente para lidar

toda a gama de jitter da rede. Na prática, um atraso de inicialização de 10-15 segundos é normal-
é adequado, portanto, este não é um grande problema.

A outra diferença importante é que os eventos de transmissão ao vivo geralmente têm dreds ou milhares de visualizadores simultâneos do mesmo conteúdo. Sob estes cir-
cumstances, a solução natural para streaming ao vivo é usar multicast. Isto é
não é o caso para streaming de mídia armazenada porque os usuários normalmente fazem streaming

conteúdo diferente em um determinado momento. O streaming para muitos usuários consiste em muitos

sessões de streaming individuais que ocorrem ao mesmo tempo.

Um esquema de streaming multicast funciona da seguinte maneira. O servidor envia cada mídia pacote uma vez usando multicast IP para um endereço de grupo. A rede fornece uma cópia de o pacote para cada membro do grupo. Todos os clientes que desejam receber o stream juntou-se ao grupo. Os clientes fazem isso usando IGMP, em vez de enviar uma mensagem RTSP para o servidor de mídia. Isso ocorre porque o servidor de mídia já está enviar a transmissão ao vivo (exceto antes de o primeiro usuário entrar). O que é necessário é providencie para que o fluxo seja recebido localmente.

Visto que multicast é um serviço de entrega um para muitos, a mídia é transportada em RTP pacotes em um transporte UDP. TCP opera apenas entre um único remetente e um único receptor. Uma vez que o UDP não fornece confiabilidade, alguns pacotes podem ser perdido. Para reduzir o nível de perda de mídia a um nível aceitável, podemos usar FEC e intercalação, como antes.

No caso do FEC, há uma interação benéfica com multicast que é mostrado no exemplo de paridade da Figura 7.56. Quando os pacotes são multicast, diferentes clientes diferentes podem perder pacotes diferentes. Por exemplo, o cliente 1 perdeu o pacote B , o cliente 2 perdeu o pacote de paridade P , o cliente 3 perdeu D e o cliente 4 não perdeu nenhum pacote

ets. No entanto, mesmo que três pacotes diferentes sejam perdidos entre os clientes, cada cliente pode recuperar todos os pacotes de dados neste exemplo. Tudo o que é necessário é que cada cliente perca no máximo um pacote, seja ele qual for, para que o pacote ausente pode ser recuperado por um cálculo de paridade. Nonnenmacher et al. (1997) descreve como essa ideia pode ser usada para aumentar a confiabilidade.

Para um servidor com um grande número de clientes, multicast de mídia em RTP e Os pacotes UDP são claramente a maneira mais eficiente de operar. Caso contrário, o servidor deve transmitir N fluxos quando tiver N clientes, o que exigirá uma quantidade de largura de banda da rede no servidor para grandes eventos de streaming.

Você pode ficar surpreso ao saber que a Internet não funciona assim na prática. O que geralmente acontece é que cada usuário estabelece um protocolo TCP separado conexão com o servidor e a mídia é transmitida por meio dessa conexão. Para o cliente, é o mesmo que streaming de mídia armazenada. E como acontece com o streaming armazenado mídia, há várias razões para essa escolha aparentemente ruim.

A primeira razão é que o multicast IP não está amplamente disponível na Internet.

Alguns ISPs e redes oferecem suporte interno, mas geralmente não está disponível em limites de rede conforme necessário para streaming de área ampla. As outras razões são

C
D
P
UMA
B
C
D
P
UMA
B
C
D
P

Figura 7-56. Mídia de streaming multicast com um pacote de paridade.

as mesmas vantagens do TCP sobre o UDP, conforme discutido anteriormente. Streaming com TCP alcançará quase todos os clientes na Internet, especialmente quando disfarçado como HTTP passar por firewalls e a entrega confiável de mídia permite que os usuários retrocedam facilmente ly.

Há um caso importante em que UDP e multicast podem ser usados para streaming, no entanto: dentro de uma rede de provedor. Por exemplo, uma empresa de cabo pode decidir transmitir canais de TV para set-top boxes de clientes usando tecnologia IP tecnologia em vez de transmissões de vídeo tradicionais. O uso de IP para distribuir amplamente o vídeo elenco é amplamente denominado IPTV, conforme discutido acima. Desde a empresa de cabo

tem controle total de sua própria rede, pode projetá-la para suportar multicast IP e ter largura de banda suficiente para distribuição baseada em UDP. Tudo isso é invisível para o cliente, visto que a tecnologia IP existe dentro do **jardim murado** do provider. Parece a TV a cabo em termos de serviço, mas é IP por baixo, com o set-top box sendo um computador executando UDP e a TV sendo simplesmente um monitor conectado ao computador.

De volta ao caso da Internet, a desvantagem da transmissão ao vivo sobre TCP é que o servidor deve enviar uma cópia separada da mídia para cada cliente. Isso é viável para um número moderado de clientes, especialmente para áudio. O truque é colocar o servidor em um local com boa conectividade com a Internet para que haja banda suficiente largura. Normalmente, isso significa alugar um servidor em um data center de um provedor de hospedagem

vidor, não usando um servidor em casa apenas com conectividade de banda larga à Internet. Existe um mercado de hospedagem muito competitivo, então isso não precisa ser caro.

Na verdade, é fácil para qualquer pessoa, até mesmo um estudante, configurar e operar um fluxo servidor de mídia, como uma estação de rádio da Internet. Os principais componentes deste

Página 748

724

A CAMADA DE APLICAÇÃO INDIVÍDUO. 7

estação são ilustradas na Figura 7-57. A base da estação é um PC comum com uma placa de som e um microfone decentes. Um software popular é usado para capturar áudio e codificá-lo em vários formatos, por exemplo, MP4, e os players de mídia são usados para ouça o áudio normalmente.

PC do aluno
Microfone
meios de comunicação
jogador
Áudio
plug-in de captura
Codec
plugar
Conexões TCP
para ouvintes
meios de comunicação
servidor
Internet

Figura 7-57. Uma estação de rádio estudantil.

O fluxo de áudio capturado no PC é então alimentado pela Internet para uma mídia servidor com boa conectividade de rede, seja como podcasts para streaming de arquivos armazenados

ou para transmissão ao vivo. O servidor lida com a tarefa de distribuição da mídia via grande número de conexões TCP. Ele também apresenta um site front-end com páginas sobre a estação e links para o conteúdo que está disponível para streaming. Existem pacotes de software comerciais para gerenciar todas as peças, bem como pacotes de código aberto, como o icecast.

No entanto, para um grande número de clientes, torna-se inviável o uso de TCP para enviar mídia para cada cliente de um único servidor. Simplesmente não há o suficiente largura de banda para um servidor. Para grandes sites de streaming, o streaming é feito usando um conjunto de servidores que estão espalhados geograficamente, para que um cliente possa se conectar

o servidor mais próximo. Esta é uma rede de distribuição de conteúdo que estudaremos no fim do capítulo.

7.4.5 Conferência em Tempo Real

Era uma vez, as chamadas de voz eram transportadas pelo telefone público comutado rede, e o tráfego da rede era principalmente tráfego de voz, com um pouco de dados tráfego aqui e ali. Então veio a Internet e a Web. O tráfego de dados cresceu e cresceu, até 1999 havia tanto tráfego de dados quanto tráfego de voz (já que voz agora está digitalizado, ambos podem ser medidos em bits). Em 2002, o volume de tráfego de dados ficava uma ordem de magnitude maior do que o volume de tráfego de voz e ainda crescendo exponencialmente, com o tráfego de voz ficando quase estável.

A consequência desse crescimento foi virar a rede telefônica em seu cabeça. O tráfego de voz agora é transportado usando tecnologias da Internet e representa apenas

Página 749

SEC. 7.4

TRANSMISSÃO DE ÁUDIO E VÍDEO

725

uma pequena fração da largura de banda da rede. Esta tecnologia disruptiva é conhecida como **voz sobre IP** e também como **telefonia via Internet**.

Voice-over-IP é usado em várias formas que são impulsionadas por fortes fatores econômicos tor. (Tradução em inglês: economiza dinheiro, então as pessoas usam.) Uma forma é ter o que parecem telefones normais (antiquados?) que se conectam à Ethernet e enviar chamadas pela rede. Pehr Anderson era um estudante de graduação em MIT quando ele e seus amigos prototiparam este design para um projeto de classe. Eles obteve uma nota "B". Não satisfeita, ele abriu uma empresa chamada NBX em 1996, foi pioneira neste tipo de voz sobre IP e vendeu-a para a 3Com por \$ 90 milhões três anos depois. As empresas adoram essa abordagem porque permite que elas acabem com o separar linhas telefônicas e se contentar com as redes que já possuem.

Outra abordagem é usar a tecnologia IP para construir um telefone de longa distância rede. Em países como os EUA, esta rede pode ser acessada por concorrentes serviço de longa distância, discando um prefixo especial. Amostras de voz são colocadas em pacotes que são injetados na rede e retirados dos pacotes quando eles deixar. Uma vez que o equipamento IP é muito mais barato do que o equipamento de telecomunicações isso leva a serviços mais baratos.

À parte, a diferença de preço não é totalmente técnica. Por muitos dez-ades, o serviço telefônico era um monopólio regulamentado que garantia a comunicação telefônica às empresas um lucro percentual fixo sobre seus custos. Não surpreendentemente, isso os levou a aumentar os custos, por exemplo, por ter muitos e muitos hardwares redundantes, justificado em nome de uma melhor confiabilidade (o sistema telefônico só foi autorizado a ser para um total de 2 horas a cada 40 anos, ou 3 min / ano em média). Este efeito era frequentemente referida como a "síndrome do poste de telefone banhado a ouro". uilação, o efeito diminuiu, é claro, mas o equipamento legado ainda existe. A indústria de TI nunca teve um histórico de operação assim, então sempre foi enxuta e significa.

No entanto, vamos nos concentrar na forma de voz sobre IP que é provavelmente o mais visível para os usuários: usando um computador para ligar para outro computador. Essa forma

tornou-se comum quando os PCs começaram a ser vendidos com microfones, alto-falantes, câmeras eras e CPUs rápidas o suficiente para processar mídia, e as pessoas começaram a se conectar ao Internet em casa com taxas de banda larga. Um exemplo bem conhecido é o software Skype ware que foi lançado a partir de 2003. Skype e outras empresas também fornecem gateways para tornar mais fácil ligar para números de telefone regulares, bem como para computadores com endereços IP.

Conforme a largura de banda da rede aumentou, as chamadas de vídeo se juntaram às chamadas de voz. Inicialmente, as videochamadas eram do domínio das empresas. Sistemas de videoconferência foram projetado para trocar vídeo entre dois ou mais locais, capacitando executivos em locais diferentes para se verem enquanto realizavam suas reuniões. Contudo, com boa conectividade de banda larga com a Internet e software de compressão de vídeo, casa os usuários também podem fazer videoconferência. Ferramentas como o Skype, que começaram como apenas áudio agora inclui rotineiramente vídeo com as chamadas para que amigos e familiares em todo o mundo pode ver e ouvir um ao outro.

Página 750

726

A CAMADA DE APLICAÇÃO INDIVÍDUO. 7

Do nosso ponto de vista, as chamadas de voz ou vídeo pela Internet também são um fluxo de mídia problema, mas é muito mais restrito do que fazer streaming de um arquivo armazenado ou um evento ao vivo. A restrição adicionada é a baixa latência necessária para um sistema bidirecional conversação. A rede telefônica permite uma latência unilateral de até 150 msec para uso aceitável, após o qual o atraso começa a ser percebido como irritante pelo participantes. (As chamadas internacionais podem ter uma latência de até 400 msec, na qual apontam que estão longe de ser uma experiência positiva do usuário.)

Essa baixa latência é difícil de alcançar. Certamente, armazenando em buffer 5-10 segundos de a mídia não vai funcionar (como funcionaria para a transmissão de um evento esportivo ao vivo). No-

em vez disso, os sistemas de vídeo e voz sobre IP devem ser projetados com uma variedade de tecnologias

técnicas para minimizar a latência. Este objetivo significa começar com UDP como a escolha clara em vez de TCP, porque as retransmissões TCP apresentam pelo menos uma viagem de ida e volta pena de atraso. Algumas formas de latência não podem ser reduzidas, no entanto, mesmo com UDP. Por exemplo, a distância entre Seattle e Amsterdã é próxima a 8.000 km. O atraso de propagação da velocidade da luz para esta distância na fibra óptica é 40 msec. Boa sorte vencendo isso. Na prática, o atraso de propagação através do rede será mais longa porque vai cobrir uma distância maior (os bits não seguem baixo uma rota de grande círculo) e têm atrasos de transmissão conforme cada roteador IP armazena e

encaminha um pacote. Esse atraso fixo corrói o orçamento de atraso aceitável.

Outra fonte de latência está relacionada ao tamanho do pacote. Normalmente, pacotes grandes são a melhor maneira de usar a largura de banda da rede porque são mais eficientes. Como sempre, a uma taxa de amostragem de áudio de 64 kbps, um pacote de 1 KB levaria 125 ms para preencher (e ainda mais se as amostras forem comprimidas). Este atraso consumiria a maior parte do orçamento geral para atrasos. Além disso, se o pacote de 1 KB for enviado por um link de acesso de banda larga que funciona a apenas 1 Mbps, levará 8 ms para transmitir. Em seguida, adicione mais 8 msec para o pacote passar pelo link de banda larga no outro fim. Claramente, pacotes grandes não funcionarão.

Em vez disso, os sistemas de voz sobre IP usam pacotes curtos para reduzir a latência ao custo da eficiência da largura de banda. Eles agrupam amostras de áudio em unidades menores, geralmente 20 msec. A 64 kbps, são 160 bytes de dados, menos com a compactação. Contudo, por definição, o atraso desse empacotamento será de 20 ms. A transmissão o retardamento também será menor porque o pacote é mais curto. Em nosso exemplo,

reduziria para cerca de 1 mseg. Usando pacotes curtos, o mínimo de sentido único o atraso de um pacote de Seattle para Amsterdã foi reduzido de um inaceitável 181 mseg ($40 + 125 + 16$) a 62 mseg ($40 + 20 + 2$) aceitáveis.

Ainda não falamos sobre a sobrecarga do software, mas também vai consumir parte do orçamento de atraso. Isso é especialmente verdadeiro para vídeo, uma vez que a compressão é geralmente necessário para ajustar o vídeo à largura de banda disponível. Ao contrário do streaming de um arquivo armazenado, não há tempo para ter um codificador de computação intensiva para alta níveis de compressão. O codificador e o decodificador devem ser executados rapidamente. O armazenamento em buffer ainda é necessário para reproduzir as amostras de mídia a tempo (para evitar áudio ou vídeo irregular), mas a quantidade de buffer deve ser mantida muito pequeno, pois o tempo restante em nosso orçamento de atraso é medido em milissegundos.

Página 751

SEC. 7,4
TRANSMISSÃO DE ÁUDIO E VÍDEO

727

Quando um pacote demora muito para chegar, o jogador pula o mesmo, por exemplo, talvez reproduzindo ruído ambiente ou repetindo um quadro para mascarar a perda para o utilizador. Há uma compensação entre o tamanho do buffer usado para lidar com jitter e a quantidade de mídia perdida. Um buffer menor reduz a latência, mas resulta em mais perda devido ao jitter. Eventualmente, conforme o tamanho do buffer diminui, a perda tornam-se perceptíveis para o usuário.

Leitores observadores podem ter notado que não dissemos nada sobre a rede protocolos da camada de trabalho até agora nesta seção. A rede pode reduzir a latência, ou menos jitter, usando mecanismos de qualidade de serviço. O motivo desse problema ter não surgiu antes é que o streaming é capaz de operar com latência substancial, mesmo no caso de streaming ao vivo. Se a latência não for uma grande preocupação, um buffer no host final é suficiente para lidar com o problema de jitter. No entanto, para condições em tempo real

referência, geralmente é importante que a rede reduza o atraso e o jitter para ajudar a cumprir o orçamento de atraso. O único momento em que não é importante é quando há tanta largura de banda de rede que todos recebem um bom serviço.

No cap. 5, descrevemos dois mecanismos de qualidade de serviço que ajudam com este objetivo. Um mecanismo é DS (Differentiated Services), em que os pacotes são marcados como pertencentes a classes diferentes que recebem tratamento diferente dentro do rede. A marcação apropriada para pacotes de voz sobre IP é baixo atraso. Na prática, os sistemas definem o ponto de código DS para o valor conhecido para o *Expedited Classe de encaminhamento* com tipo de serviço *Low Delay*. Isso é especialmente útil durante links de acesso de banda larga, pois esses links tendem a ficar congestionados quando o tráfego da Web ou

outro tráfego compete pelo uso do link. Dado um caminho de rede estável, atraso e o jitter é aumentado pelo congestionamento. Cada pacote de 1 KB leva 8 ms para ser enviado por um Link de 1 Mbps, e um pacote de voz sobre IP incorrerá nesses atrasos se estiver em um fila atrás do tráfego da Web. No entanto, com um baixo atraso marcando a voz sobre IP os pacotes saltarão para o topo da fila, ignorando os pacotes da Web e o atraso deles.

O segundo mecanismo que pode reduzir o atraso é certificar-se de que há suficiente largura de banda fiável. Se a largura de banda disponível varia ou a taxa de transmissão fluctua (como com vídeo compactado) e às vezes não há banda suficiente largura, as filas aumentarão e aumentarão o atraso. Isso ocorrerá mesmo com o DS. Para garantir largura de banda suficiente, uma reserva pode ser feita com a rede. Isto capacidade é fornecida por serviços integrados. Infelizmente, não é amplamente planejado. Em vez disso, as redes são projetadas para um nível de tráfego esperado ou rede

os clientes recebem acordos de nível de serviço para um determinado nível de tráfego. Os aplicativos devem operar abaixo deste nível para evitar causar congestionamento e introdução reduzindo atrasos desnecessários. Para videoconferências casuais em casa, o usuário pode escolha uma qualidade de vídeo como proxy para as necessidades de largura de banda, ou o software pode testar o caminho da rede e selecione uma qualidade apropriada automaticamente. Qualquer um dos fatores acima pode fazer com que a latência se torne inaceitável, então a conferência em tempo real requer que se preste atenção a todos eles. Por mais visão de voz sobre IP e análise desses fatores, ver Goode (2002).

Página 752

728

A CAMADA DE APLICAÇÃO INDIVÍDUO. 7

Agora que discutimos o problema de latência no streaming de mídia caminho, passaremos para o outro problema principal que os sistemas de conferência devem endereço. Esse problema é como configurar e desativar chamadas. Vamos olhar para dois protocolos amplamente utilizados para esse fim, H.323 e SIP. Skype é outra sistema importante, mas seu funcionamento interno é proprietário.

H.323

Uma coisa que estava clara para todos antes das chamadas de voz e vídeo serem feitas pela Internet era que, se cada fornecedor projetasse sua própria pilha de protocolo, o sistema tem nunca funcionaria. Para evitar esse problema, várias partes interessadas conseguiram juntos sob os auspícios da UIT para elaborar padrões. Em 1996, a ITU emitiu uma recomendação recomendação **H.323**, intitulada " Visual Telephone Systems and Equipment for Local Redes de área que fornecem qualidade de serviço não garantida. " Apenas a indústria de telefonia pensaria em tal nome. Foi rapidamente alterado para " Pack-et-based Multimedia Communications Systems " na revisão de 1998. H.323 era a base para os primeiros sistemas de conferência pela Internet amplamente difundidos. Permanece o solução mais amplamente implantada, em sua sétima versão em 2009.

H.323 é mais uma visão geral arquitetônica da telefonia da Internet do que uma espécie protocolo específico. Ele faz referência a um grande número de protocolos específicos para códigos de fala , configuração de chamadas, sinalização, transporte de dados e outras áreas, em vez de especificar essas coisas em si. O modelo geral é ilustrado na Figura 7.58. No centro está um **gateway** que conecta a Internet à rede telefônica. Fala o H.323 protocolos no lado da Internet e os protocolos PSTN no lado do telefone. os dispositivos de comunicação são chamados de **terminais**. Uma LAN pode ter um **gatekeeper** , que controla os pontos finais sob sua jurisdição, chamados de **zona** .

Internet
Gatekeeper
Telefone
rede
Zona
terminal
Porta de entrada

Figura 7-58. O modelo de arquitetura H.323 para telefonia pela Internet.

Uma rede telefônica precisa de vários protocolos. Para começar, há um protocolo para codificação e decodificação de áudio e vídeo. Representantes de telefonia padrão entações de um único canal de voz como 64 kbps de áudio digital (8000 amostras de 8 bits por segundo) são definidos na recomendação **G.711** da ITU . Todos os sistemas H.323

Página 753

SEC. 7,4

TRANSMISSÃO DE ÁUDIO E VÍDEO

729

deve suportar G.711. Outras codificações que comprimem a fala são permitidas, mas não requeridos. Eles usam algoritmos de compressão diferentes e fazem negócios diferentes offs entre qualidade e largura de banda. Para vídeo, as formas MPEG de vídeo com-

pressão que descrevemos acima são suportados, incluindo H.264. Uma vez que múltiplos algoritmos de compressão são permitidos, um protocolo é necessário para permitem que os terminais negociem qual usuário. Este protocolo é chamado **H.245**. Ele também negocia outros aspectos da conexão, como o bit taxa. O RTCP é necessário para o controle dos canais RTP. Também é necessário um protocolo para estabelecer e liberar conexões, fornecendo tons de discagem, tornando sons e o resto da telefonia padrão. ITU **Q.931** é usado aqui. Os terminais também precisam de um protocolo para falar com o gatekeeper (se houver). Para esse propósito, o **H.225** é usado. O canal de PC para gatekeeper que gerencia é chamado o **RAS (Registro / Admissão / status)** do canal. Este canal permite terminais para entrar e sair da zona, solicitar e retornar largura de banda e fornecer o status atualizações, entre outras coisas. Finalmente, um protocolo é necessário para os dados reais transmissão. RTP sobre UDP é usado para este propósito. É gerenciado por RTCP, como de costume. O posicionamento de todos esses protocolos é mostrado na Figura 7.59.

Protocolo de camada de enlace
 IP
 Áudio
 G.7xx
 RTP
 Protocolo da camada física
 TCP
 UDP
 Vídeo
 H.26x
 RTCP
 H.225
 (RAS)
 Q.931
 (Sinalização)
 H.245
 (Ligar
 Ao controle)
 Ao controle

Figura 7-59. A pilha do protocolo H.323.

Para ver como esses protocolos se encaixam, considere o caso de um terminal de PC em uma LAN (com um gatekeeper) chamando um telefone remoto. O PC primeiro tem que descobrir o gatekeeper, então ele transmite um pacote de descoberta do gatekeeper UDP para a porta 1718. Quando o gatekeeper responde, o PC aprende o endereço IP do gatekeeper. Agora o PC se registra com o gatekeeper enviando uma mensagem RAS em um UDP pacote. Depois de ser aceito, o PC envia ao gatekeeper uma admissão RAS mensagem solicitando largura de banda. Somente depois que a largura de banda for concedida pode chamar a configuração começar. A ideia de solicitar largura de banda com antecedência é permitir que o portão detentor para limitar o número de chamadas. Isso pode evitar o excesso de assinaturas indo em linha para ajudar a fornecer a qualidade de serviço necessária.

730

A CAMADA DE APLICAÇÃO INDIVÍDUO. 7

À parte, o sistema telefônico faz a mesma coisa. Quando você pega o receptor, um sinal é enviado para a estação final local. Se o escritório tiver capacidade para outra chamada, ele gera um tom de discagem. Se não, você não ouve nada. Nowadays, o sistema é tão superdimensionado que o tom de discagem é quase sempre instantâneo, mas nos primeiros dias da telefonia, geralmente demorava alguns segundos. Então seus netos sempre perguntam " Por que existem tons de discagem? " Agora você sabe. Excepto então, provavelmente os telefones não existirão mais.

O PC agora estabelece uma conexão TCP com o gatekeeper para iniciar a chamada configuração. A configuração de chamadas usa protocolos de rede telefônica existentes, que são conexão orientado, então o TCP é necessário. Em contraste, o sistema telefônico não tem nada como RAS para permitir que os telefones anunciem sua presença, então os designers do H.323 foram

livre para usar UDP ou TCP para RAS e escolheram o UDP de menor sobrecarga. Agora que tem largura de banda alocada, o PC pode enviar uma mensagem de *CONFIGURAÇÃO* Q.931 sage sobre a conexão TCP. Esta mensagem especifica o número do telefone sendo chamado (ou o endereço IP e a porta, se um computador estiver sendo chamado). o gatekeeper responde com uma mensagem *CALL PROCEEDING* Q.931 para confirmar recebimento correto da solicitação. O gatekeeper então encaminha a mensagem *SETUP* para o gateway.

O gateway, que é meio computador, meio interruptor de telefone, faz um chamada telefônica comum para o telefone (comum) desejado. O escritório final para ao qual o telefone está conectado toca o telefone chamado e também envia de volta um Mensagem de *ALERTA* Q.931 para informar ao PC chamador que o toque começou. Quando o a pessoa do outro lado pega o telefone, a estação final envia de volta um Q.931 Mensagem *CONNECT* para sinalizar ao PC que há uma conexão.

Uma vez que a conexão foi estabelecida, o gatekeeper não está mais no loop, embora o gateway seja, é claro. Os pacotes subsequentes contornam o portão protetor e vão diretamente para o endereço IP do gateway. Neste ponto, temos apenas um tubo vazio correndo entre as duas partes. Esta é apenas uma condição da camada física conexão para mover bits, nada mais. Nenhum lado sabe nada sobre o outro 1.

O protocolo H.245 agora é usado para negociar os parâmetros da chamada. isto usa o canal de controle H.245, que está sempre aberto. Cada lado começa por anunciar suas capacidades, por exemplo, se ele pode lidar com vídeo (H.323 pode lidar com vídeo) ou chamadas em conferência, quais codecs ele suporta, etc. Uma vez de cada lado sabe o que o outro pode controlar, dois canais de dados unidirecionais são configurados e um codec e outros parâmetros são atribuídos a cada um. Uma vez que cada lado pode têm equipamentos diferentes, é inteiramente possível que os codecs à frente e canais reversos são diferentes. Depois que todas as negociações forem concluídas, o fluxo de dados pode começar a usar RTP. É gerenciado usando RTCP, que desempenha um papel no congestionamento ao controle. Se houver vídeo, o RTCP trata da sincronização de áudio / vídeo. os vários canais são mostrados na Figura 7.60. Quando qualquer uma das partes desliga, o Q.931 canal de sinalização de chamada é usado para interromper a conexão após a chamada ter sido concluída para liberar recursos que não são mais necessários.

Página 755

SEC. 7,4
TRANSMISSÃO DE ÁUDIO E VÍDEO

731

Canal de controle de dados (RTCP)
Canal reverso de dados (RTP)
Canal de transmissão de dados (RTP)
Canal de controle de chamada (H.245)
Canal de sinalização de chamadas (Q.931)
Chamador
Callee

Figura 7-60. Canais lógicos entre o chamador e o receptor durante uma chamada.

Quando a chamada é encerrada, o PC chamador contata o gatekeeper novamente com uma mensagem RAS para liberar a largura de banda atribuída. Alternativamente, pode fazer outra chamada.

Não dissemos nada sobre a qualidade do serviço como parte do H.323, mesmo embora tenhamos dito que é uma parte importante de tornar a conferência em tempo real um sucesso. O motivo é que o QoS está fora do escopo do H.323. Se o underlying-rede de conexão é capaz de produzir uma conexão estável e sem jitter a partir da rede conectando o PC ao gateway, a QoS na chamada será boa; caso contrário, não será. No entanto, qualquer parte da chamada no lado do telefone será livre de instabilidade, porque é assim que a rede telefônica é projetada.

SIP - O Protocolo de Iniciação de Sessão

H.323 foi projetado pela ITU. Muitas pessoas na comunidade da Internet viram

como um produto típico de telecomunicações: grande, complexo e inflexível. Consequentemente, IETF definiu formar um comitê para projetar uma maneira mais simples e modular de fazer voz sobre IP. O principal resultado até agora é o **SIP** (**S**ession **I**nitiati**O**n **P**rotocol). A última versão é descrito no RFC 3261, que foi escrito em 2002. Este protocolo descreve como configurar chamadas telefônicas pela Internet, videoconferências e outros recursos multimídia conexões. Ao contrário do H.323, que é um pacote de protocolo completo, o SIP é um único módulo, mas foi projetado para funcionar bem com aplicativos de Internet existentes ções. Por exemplo, define números de telefone como URLs, para que as páginas da Web possam contê-los, permitindo um clique em um link para iniciar uma chamada telefônica (da mesma forma o esquema *mailto* permite um clique em um link para abrir um programa para enviar um e-mail mensagem). O SIP pode estabelecer sessões de duas partes (chamadas telefônicas comuns), multipartidário sessões (onde todos podem ouvir e falar) e sessões multicast (um remetente, muitos receptores). As sessões podem conter áudio, vídeo ou dados, sendo o último útil para jogos multiplayer em tempo real, por exemplo. SIP apenas lida com a configuração, cara encerramento e encerramento das sessões. Outros protocolos, como RTP / RTCP, são

Página 756

732

A CAMADA DE APLICAÇÃO INDIVÍDUO. 7

também usado para transporte de dados. SIP é um protocolo da camada de aplicação e pode ser executado sobre UDP ou TCP, conforme necessário.

O SIP oferece suporte a uma variedade de serviços, incluindo localização do receptor (que não pode estar em sua máquina de casa) e determinar as capacidades do receptor, bem como han-controlando a mecânica de configuração e encerramento de chamadas. No caso mais simples, conjuntos SIP

uma sessão do computador do chamador para o computador do receptor, então vamos examinar ine esse caso primeiro.

Os números de telefone no SIP são representados como URLs usando o esquema *sip* , para exemplo, *sip: ilse@cs.university.edu* para um usuário chamado Ilse no host especificado por o nome DNS *cs.university.edu* . URLs SIP também podem conter endereços IPv4, Endereços IPv6 ou números de telefone reais.

O protocolo SIP é um protocolo baseado em texto modelado em HTTP. Uma parte envia uma mensagem em texto ASCII que consiste em um nome de método na primeira linha, seguido por linhas adicionais contendo cabeçalhos para passar parâmetros. Muitos dos principais ers são retirados do MIME para permitir que o SIP interaja com os aplicativos de Internet existentes ções. Os seis métodos definidos pela especificação principal estão listados na Figura 7.61.

Método

Descrição

CONVITE

Solicitar o início de uma sessão

ACK

Confirme se uma sessão foi iniciada

TCHAU

Solicitar o encerramento de uma sessão

OPÇÕES

Consulte um host sobre seus recursos

CANCELAR

Cancelar um pedido pendente

REGISTRO

Informe um servidor de redirecionamento sobre a localização atual do usuário

Figura 7-61. Métodos SIP.

Para estabelecer uma sessão, o chamador cria uma conexão TCP com a chamada lee e envia uma mensagem *INVITE* sobre ele ou envia a mensagem *INVITE* em um UDP pacote. Em ambos os casos, os cabeçalhos na segunda linha e nas linhas subsequentes descrevem o estrutura do corpo da mensagem, que contém os recursos do chamador, mídia tipos e formatos. Se o receptor aceitar a chamada, ele responde com um tipo de HTTP

código de resposta (um número de três dígitos usando os grupos da Fig.7-38, 200 para aceitação). Seguindo a linha do código de resposta, o receptor também pode fornecer informações sobre seus recursos, tipos de mídia e formatos.

A conexão é feita usando um handshake de três vias, para que o chamador responda com uma mensagem *ACK* para finalizar o protocolo e confirmar o recebimento da mensagem 200. Qualquer uma das partes pode solicitar o encerramento de uma sessão, enviando uma mensagem com

o método *BYE*. Quando o outro lado reconhece, a sessão é encerrada.

O método *OPTIONS* é usado para consultar uma máquina sobre seus próprios recursos.

É normalmente usado antes de uma sessão ser iniciada para descobrir se essa máquina é uniforme capaz de voz sobre IP ou qualquer tipo de sessão que está sendo contemplada.

Página 757

SEC. 7,4

TRANSMISSÃO DE ÁUDIO E VÍDEO

733

O método *REGISTER* está relacionado à capacidade do SIP de rastrear e se conectar a um usuário que está fora de casa. Esta mensagem é enviada para um servidor de localização SIP que mantém o controle de quem está onde. Esse servidor pode ser posteriormente consultado para encontrar o usuário

localização atual. A operação de redirecionamento é ilustrada na Figura 7.62. Aqui o chamador envia a mensagem *INVITE* para um servidor proxy para ocultar o possível redirecionamento.

O proxy então procura onde o usuário está e envia a mensagem *INVITE* para lá. Isto em seguida, atua como um retransmissor para as mensagens subsequentes no handshake triplo. As mensagens *LOOKUP* e *REPLY* não fazem parte do SIP; qualquer protocolo conveniente pode ser usado, dependendo do tipo de servidor de localização usado.

6 OK
5 OK
1 CONVITE
2
OLHO PARA CIMA
3
RESPOSTA
4 CONVIDAR
7 ACK
8 ACK
Chamador
Callee
Servidor de localização
Proxy
9 dados

Figura 7-62. Utilização de servidor proxy e redirecionamento com SIP.

O SIP tem uma variedade de outros recursos que não descreveremos aqui, incluindo chamada em espera, filtragem de chamadas, criptografia e autenticação. Ele também tem a habilidade

para fazer chamadas de um computador para um telefone comum, se um gateway adequado entre a Internet e o sistema telefônico está disponível.

Comparação de H.323 e SIP

Tanto o H.323 quanto o SIP permitem chamadas de duas partes e multipartes usando ambos computadores e telefones como terminais. Ambos suportam negociação de parâmetros, criptografia e os protocolos RTP / RTCP. Um resumo de suas semelhanças e diferenças referências são apresentadas na Figura 7.63.

Embora os conjuntos de recursos sejam semelhantes, os dois protocolos diferem amplamente na filosofia

Sophy. H.323 é um padrão típico e pesado da indústria de telefonia, especificando a pilha de protocolo completa e definindo precisamente o que é permitido e o que é proibido. Esta abordagem leva a protocolos muito bem definidos em cada camada, facilitando a tarefa de interoperabilidade. O preço pago é um padrão grande, complexo e rígido que é difícil de se adaptar a aplicações futuras.

Em contraste, o SIP é um protocolo típico da Internet que funciona trocando linhas de texto ASCII. É um módulo leve que funciona bem com outros In-

protocolos ternet, mas não tão bem com os protocolos de sinalização de sistema telefônico existentes.

734

A CAMADA DE APLICAÇÃO
INDIVÍDUO. 7

Item

H.323

trago

Projetado por

ITU

IETF

Compatibilidade com PSTN

sim

Largamente

Compatibilidade com Internet Sim, ao longo do tempo

sim

Arquitetura

Monolítico

Modular

Integridade

Pilha de protocolo completa

SIP apenas lida com a configuração

Negociação de parâmetros

sim

sim

Sinalização de chamada

Q.931 sobre TCP

SIP sobre TCP ou UDP

Formato de mensagem

Binário

ASCII

Transporte de mídia

RTP / RTCP

RTP / RTCP

Chamadas multipartidárias

sim

sim

Conferências multimídia

sim

Não

Endereçando

URL ou número de telefone

URL

Terminação de chamada

Liberação explícita ou TCP

Explícito ou tempo limite

Mensagem instantânea

Não

sim

Encriptação

sim

sim

Tamanho dos padrões

1400 páginas

250 páginas

Implementação

Grande e complexo

Moderado, mas com problemas

Status

Difundido, esp. vídeo

Alternativa, esp. voz

Figura 7-63. Comparação de H.323 e SIP.

Porque o modelo IETF de voz sobre IP é altamente modular, é flexível e pode ser facilmente adaptado a novas aplicações. A desvantagem é que ele sofreu de problemas contínuos de interoperabilidade enquanto as pessoas tentam interpretar o que o padrão significa.

7.5 ENTREGA DE CONTEÚDO

A Internet costumava ser apenas comunicação, como a rede telefônica.

No início, os acadêmicos se comunicavam com máquinas remotas, conectando-se a rede para realizar tarefas. As pessoas usaram e-mail para se comunicar com cada um outro por um longo tempo, e agora usa vídeo e voz sobre IP também. Desde o A Web cresceu, no entanto, a Internet tornou-se mais voltada para o conteúdo do que para a comunicação

comunicação. Muitas pessoas usam a web para encontrar informações, e há um tremendo quantidade de compartilhamento de arquivos ponto a ponto que é impulsionado pelo acesso a filmes, músicas e programas. A mudança para o conteúdo foi tão pronunciada que a maioria dos A largura de banda ternet agora é usada para fornecer vídeos armazenados.

Página 759

SEC. 7,5
ENTREGA DE CONTEÚDO

735

Porque a tarefa de distribuição de conteúdo é diferente daquela de comunicação ção, ele coloca requisitos diferentes na rede. Por exemplo, se Sally quiser para falar com Jitu, ela pode fazer uma chamada de voz sobre IP para o celular dele. A comunicação ção deve ser com um determinado computador; não vai adiantar nada chamar a comissão de Paul puter. Mas se Jitu quiser assistir à última partida de críquete de seu time, ele ficará feliz em transmitir vídeo de qualquer computador que possa fornecer o serviço. Ele não se importa se o computador é de Sally ou Paul, ou, mais provavelmente, um servidor desconhecido em a Internet. Ou seja, a localização não importa para o conteúdo, exceto quando a feta força (e legalidade).

A outra diferença é que alguns sites que fornecem conteúdo se tornaram tremendamente popular. O YouTube é um excelente exemplo. Ele permite que os usuários compartilhem

vídeos de sua própria criação em todos os tópicos imagináveis. Muitas pessoas querem fazer esta. O resto de nós quer assistir. Com todos esses vídeos que exigem muita largura de banda, estima-se que o YouTube seja responsável por até 10% do tráfego da Internet hoje.

Nenhum servidor é poderoso ou confiável o suficiente para lidar com um nível tão surpreendente de demanda. Em vez disso, o YouTube e outros grandes provedores de conteúdo criam seus próprios redes de distribuição de conteúdo. Essas redes usam centros de dados espalhados pelo mundo para servir conteúdo a um número extremamente grande de clientes com bom desempenho desempenho e disponibilidade.

As técnicas que são usadas para distribuição de conteúdo foram desenvolvidas hora extra. No início do crescimento da Web, sua popularidade foi quase sua ruína.

Mais demandas por conteúdo levaram a servidores e redes que eram frequentemente sobrecarregado. Muitas pessoas começaram a chamar a WWW de World Wide Wait.

Em resposta à demanda do consumidor, grandes quantidades de largura de banda eram idealizada no núcleo da Internet, e uma conectividade de banda larga mais rápida foi lançada na borda da rede. Essa largura de banda foi fundamental para melhorar força, mas é apenas parte da solução. Para reduzir os atrasos intermináveis, re os pesquisadores também desenvolveram diferentes arquiteturas para usar a largura de banda para distribuição conteúdo de ing.

Uma arquitetura é uma **CDN (Rede de distribuição de conteúdo)**. Nele, um pro Vider configura uma coleção distribuída de máquinas em locais dentro da Internet e os usa para fornecer conteúdo aos clientes. Esta é a escolha dos grandes jogadores. A arquitetura alternativa é uma rede **P2P (Peer-to-Peer)**. Nele, uma coleção de os computadores reúnem seus recursos para servir conteúdo uns aos outros, sem separadamente servidores provisionados ou qualquer ponto central de controle. Esta ideia conquistou pessoas imaginação, porque, agindo juntos, muitos pequenos jogadores podem embalar um soco enorme.

Nesta seção, veremos o problema de distribuição de conteúdo no

ternet e algumas das soluções que são utilizadas na prática. Após uma breve discussão popularidade do conteúdo e tráfego da Internet, descreveremos como construir Servidores da Web e uso de cache para melhorar o desempenho dos clientes da Web. Então nós chegaremos às duas principais arquiteturas de distribuição de conteúdo: CDNs e P2P redes. Seu design e propriedades são bastante diferentes, como veremos.

Página 760

736

A CAMADA DE APLICAÇÃO
INDIVÍDUO. 7

7.5.1 Conteúdo e tráfego da Internet

Para projetar e projetar redes que funcionam bem, precisamos compreender o tráfego que eles devem transportar. Com a mudança para o conteúdo, por exemplo, servidores migraram dos escritórios da empresa para centros de dados da Internet que fornecem grandes várias máquinas com excelente conectividade de rede. Para executar mesmo um pequeno servidor hoje em dia é mais fácil e barato alugar um servidor virtual hospedado em um servidor data center líquido do que operar uma máquina real em uma casa ou escritório com banda larga conectividade com a Internet.

Felizmente, existem apenas dois fatos sobre o tráfego da Internet que são essenciais para conhecer. O primeiro fato é que muda rapidamente, não só nos detalhes, mas também no maquiagem geral. Antes de 1994, a maior parte do tráfego era transferência de arquivo FTP tradicional (para

transferência de programas e conjuntos de dados entre computadores) e e-mail. Então a Web é rivou e cresceu exponencialmente. O tráfego da web deixou o tráfego de FTP e e-mail na poeira muito antes da bolha ponto com de 2000. Começando por volta de 2000, compartilhamento de arquivos P2P para música e filmes decolaram. Em 2003, a maior parte do tráfego da Internet era tráfego P2P, deixando a Web na poeira. Em algum momento no final dos anos 2000, o vídeo transmitido usando os métodos de distribuição de conteúdo por sites como o YouTube começaram a exceder o tráfego P2P.

Em 2014, a Cisco prevê que 90% de todo o tráfego da Internet será de vídeo em uma forma ou outro (Cisco, 2010).

Nem sempre é o volume de tráfego que importa. Por exemplo, enquanto voz sobre IP o tráfego disparou antes mesmo do Skype começar em 2003, sempre será um pequeno pontinho no gráfico porque os requisitos de largura de banda de áudio são duas ordens de magnitude menor do que para o vídeo. No entanto, o tráfego de voz sobre IP sobrecarrega a rede de outras maneiras, porque é sensível à latência. Como outro exemplo, social online redes têm crescido furiosamente desde o início do Facebook em 2004. Em 2010, para o Pela primeira vez, o Facebook alcançou mais usuários na Web por dia do que o Google. Até deixando o tráfego de lado (e há uma quantidade enorme de tráfego), a rede social online trabalhos são importantes porque estão mudando a maneira como as pessoas interagem por meio de a Internet.

O que queremos dizer é que as mudanças sísmicas no tráfego da Internet acontecem rapidamente sim, e com alguma regularidade. O que virá a seguir? Verifique novamente no dia 6 edição deste livro e nós avisaremos.

O segundo fato essencial sobre o tráfego da Internet é que ele é altamente distorcido.

Muitas propriedades com as quais estamos familiarizados estão agrupadas em torno de uma média. Para

Por exemplo, a maioria dos adultos está perto da altura média. Existem algumas pessoas altas e algumas pessoas baixas, mas poucas pessoas muito altas ou muito baixas. Para esses tipos de propriedades, é possível projetar para uma faixa que não é muito grande, mas ainda assim captura a maioria da população.

O tráfego da Internet não é assim. Há muito tempo, sabe-se que havia são um pequeno número de sites com tráfego massivo e um grande número de sites site com tráfego muito menor. Este recurso tornou-se parte da linguagem de networking. Os primeiros artigos falavam sobre o tráfego em termos de **trens de pacotes**, a ideia

SEC. 7,5
ENTREGA DE CONTEÚDO

737

sendo que trens expressos com um grande número de pacotes viajariam repentinamente por um link (Jain e Routhier, 1986). Isso foi formalizado como a noção de **self-similaridade**, que para nossos propósitos pode ser considerada como tráfego de rede que exibe bits muitos intervalos curtos e muitos intervalos longos, mesmo quando vistos em escalas de tempo diferentes

(Leland et al., 1994). Trabalho posterior falou de longos fluxos de tráfego como **elefantes** e o tráfego curto fluí como **ratos**. A ideia é que existem apenas alguns elefantes e muitos ratos, mas os elefantes são importantes porque são muito grandes.

Voltando ao conteúdo da Web, o mesmo tipo de distorção é evidente. Experiência com locadoras de vídeo, bibliotecas públicas e outras organizações semelhantes mostram que não todos os itens são igualmente populares. Experimentalmente, quando N filmes estão disponíveis, o fração de todas as solicitações para o k -ésimo mais popular é de aproximadamente C/k . Aqui, C é calculado para normalizar a soma para 1, ou seja,

$$C = 1 / (1 + 1/2 + 1/3 + 1/4 + 1/5 + \dots + 1/N)$$

Assim, o filme mais popular é sete vezes mais popular que o número sete filme. Esse resultado é conhecido como **lei de Zipf** (Zipf, 1949). É nomeado após George Zipf, um professor de lingüística da Universidade Harvard que observou que a frequência do uso de uma palavra em um grande corpo de texto é inversamente proporcional à sua classificação. Para

exemplo, a 40^a palavra mais comum é usada duas vezes mais que a 80^a mais palavra comum e três vezes mais que a 120^a palavra mais comum.

Uma distribuição Zipf é mostrada na Figura 7.64 (a). Ele captura a noção de que há um pequeno número de itens populares e muitos itens impopulares. Para reconhecer nesse distribuições dessa forma, é conveniente plotar os dados em uma escala logarítmica ambos os eixos, conforme mostrado na Fig. 7-64 (b). O resultado deve ser uma linha reta.

(uma)

1

Relativo

Frequência

Classificação

0

1

5

10

15

20

Relativo

Frequência

Classificação

1

10^{-2}

10^{-1}

10^0

10^1

10^2

10^0

(b)

Figura 7-64. Distribuição Zipf (a) Em escala linear. (b) Em uma escala log-log.

Quando as pessoas olharam para a popularidade das páginas da Web, também descobriram seguem aproximadamente a lei de Zipf (Breslau et al., 1999). Uma distribuição Zipf é um exemplo em uma família de distribuições conhecidas como **leis de potência**. Leis de potência são evidentes

738

A CAMADA DE APLICAÇÃO
INDIVÍDUO. 7

em muitos fenômenos humanos, como a distribuição das populações da cidade e de riqueza. Eles têm a mesma propensão para descrever alguns jogadores grandes e um ótimo muitos jogadores menores, e eles também aparecem como uma linha reta em um gráfico de registro. isto

foi logo descoberto que a topologia da Internet poderia ser aproximadamente descrita com as leis de potência (Faloutsos et al., 1999). Em seguida, os pesquisadores começaram a traçar cada

propriedade imaginável da Internet em uma escala logarítmica, observando uma linha reta, e gritando: "Lei de potência!"

No entanto, o que importa mais do que uma linha reta em um gráfico log-log é o que essas distribuições significam para o projeto e uso de redes. Dado os muitos formas de conteúdo que têm distribuições Zipf ou power law, parece fundamental que os sites na Internet são semelhantes ao Zipf em popularidade. Isso, por sua vez, significa que um site médio não é uma representação útil. Os sites são melhor descritos como popular ou impopular. Ambos os tipos de sites são importantes. Os sites populares obviamente importante, já que alguns sites populares podem ser responsáveis pela maior parte do tráfego no Internet. Talvez surpreendentemente, os sites impopulares também podem ser importantes. Isto é porque

a quantidade total de tráfego direcionado a sites impopulares pode somar até um grande fração do tráfego geral. A razão é que existem tantos sites impopulares.

A noção de que, coletivamente, muitas escolhas impopulares podem importar tem sido popularizado por livros como *The Long Tail* (Anderson, 2008a).

Curvas mostrando decadência como a da Fig. 7-64 (a) são comuns, mas não são tudo o mesmo. Em particular, situações em que a taxa de degradação é proporcional ao quanto material resta (como átomos radioativos instáveis) exibir

decadência exponencial, que cai muito mais rápido do que a Lei de Zipf. O número de itens, digamos átomos, restantes após o tempo t é geralmente expresso como $e^{-t/a}$

, onde a constante

α determina a rapidez da decadência. A diferença entre decaimento exponencial e a Lei de Zipf é que, com decadência exponencial, é seguro ignorar o fim da cauda mas com a Lei de Zipf o peso total da cauda é significativo e não pode ser ignorado.

Para trabalhar de forma eficaz neste mundo distorcido, devemos ser capazes de construir os dois tipos

de sites. Sites impopulares são fáceis de manusear. Usando DNS, muitos sites podem, na verdade, apontar para o mesmo computador na Internet que executa todos os sites. Por outro lado, sites populares são difíceis de manusear. Não há nenhum computador, mesmo remotamente poderoso o suficiente, e usando um único computador tornar o site inacessível para milhões de usuários se ele falhar. Para lidar com esses sites, nós deve construir sistemas de distribuição de conteúdo. Vamos começar essa missão a seguir.

7.5.2 Server Farms e Web Proxies

Os designs da Web que vimos até agora têm uma única máquina de servidor falando - para várias máquinas clientes. Para construir grandes sites da Web com bom desempenho, nós pode acelerar o processamento no lado do servidor ou no lado do cliente. No servidor lado, servidores Web mais poderosos podem ser construídos com um server farm, no qual um cluster de computadores atua como um único servidor. Do lado do cliente, um melhor desempenho pode

SEC. 7,5

ENTREGA DE CONTEÚDO

739

ser alcançado com melhores técnicas de armazenamento em cache. Em particular, os caches de proxy fornecem um grande cache compartilhado para um grupo de clientes.

Descreveremos cada uma dessas técnicas separadamente. No entanto, observe que nenhuma técnica é suficiente para construir os maiores sites. Esses sites populares exigem os métodos de distribuição de conteúdo que descrevemos nas seções a seguir, que combine computadores em muitos locais diferentes.

Farms de servidores

Não importa quanta largura de banda uma máquina tenha, ela só pode servir a muitos

As solicitações da Web antes do carregamento são muito grandes. A solução neste caso é usar mais do que um computador para fazer um servidor web. Isso leva ao modelo de **farm de servidores** de Fig. 7-65.

A parte dianteira
Processo interno
base de dados
Internet
Acesso
Clientes
Servidores
Farm de servidores
Carga de saldos
entre servidores

Figura 7-65. Um farm de servidores.

A dificuldade com este modelo aparentemente simples é que o conjunto de computadores que compõem o farm de servidores deve parecer um único site lógico para os clientes. E se eles não o fazem, nós apenas configuramos diferentes sites que funcionam em paralelo.

Existem várias soluções possíveis para fazer o conjunto de servidores parecer ser um site. Todas as soluções assumem que qualquer um dos servidores pode lidar com um request de qualquer cliente. Para fazer isso, cada servidor deve ter uma cópia do site.

Os servidores são mostrados como conectados a um banco de dados back-end comum por um tracejado linha para este fim.

Uma solução é usar o DNS para espalhar as solicitações entre os servidores do servidor ver fazenda. Quando uma solicitação DNS é feita para o URL da Web, o servidor DNS retorna uma lista rotativa dos endereços IP dos servidores. Cada cliente tenta um endereço IP, normalmente o primeiro da lista. O efeito é que diferentes clientes contatam diferentes servidores para acessar o mesmo site, da maneira pretendida. O método DNS está no coração dos CDNs, e vamos revisitá-lo mais tarde nesta seção.

As outras soluções são baseadas em um **front end** que espalha as solicitações recebidas sobre o pool de servidores no farm de servidores. Isso acontece mesmo quando o cliente

Página 764

740

A CAMADA DE APLICAÇÃO

INDIVÍDUO. 7

contata o server farm usando um único endereço IP de destino. O front-end é geralmente um switch de camada de link ou um roteador IP, ou seja, um dispositivo que lida com frames

pacotes. Todas as soluções são baseadas nele (ou nos servidores) espiando na rede cabeçalhos de camada de trabalho, transporte ou aplicação e usando-os de maneiras não padronizadas.

Uma solicitação e uma resposta da Web são transportadas como uma conexão TCP. Para funcionar corretamente,

o front end deve distribuir todos os pacotes de uma solicitação para o mesmo servidor.

Um design simples é para o front-end transmitir todas as solicitações recebidas para todos os servidores. Cada servidor responde apenas a uma fração das solicitações anteriores acordo. Por exemplo, 16 servidores podem olhar para o endereço IP de origem e responder à solicitação somente se os últimos 4 bits do endereço IP de origem corresponderem aos seletores. Outros pacotes são descartados. Embora isso seja um desperdício de banda de entrada largura, muitas vezes as respostas são muito mais longas do que a solicitação, por isso não é tão ineficiente como parece.

Em um design mais geral, o front end pode inspecionar o IP, TCP e HTTP cabeçalhos de pacotes e mapeá-los arbitrariamente para um servidor. O mapeamento é chamado de política de **balanceamento de carga**, pois o objetivo é balancear a carga de trabalho entre os servidores.

A política pode ser simples ou complexa. Uma política simples pode ser usar os servidores um após o outro, por sua vez, ou round-robin. Com essa abordagem, o front end deve lembrar-se do mapeamento para cada solicitação de modo que os pacotes subsequentes que fazem parte do

a mesma solicitação será enviada para o mesmo servidor. Além disso, para tornar o site mais confiável capaz do que um único servidor, o front-end deve notar quando os servidores falharam e pare de enviar solicitações.

Muito parecido com o NAT, este projeto geral é perigoso, ou pelo menos frágil, pois nós acabaram de criar um dispositivo que viola o princípio mais básico do proto em camadas cols: cada camada deve usar seu próprio cabeçalho para fins de controle e não pode inspecionar e usar as informações da carga útil para qualquer finalidade. Mas as pessoas projetam tais sistemas de qualquer maneira e quando eles quebrarem no futuro devido a mudanças na camada superior

ers, eles tendem a se surpreender. O front-end, neste caso, é um switch ou roteador, mas ele pode agir com base nas informações da camada de transporte ou superior. Essa caixa é chamado de **middlebox** porque se interpõe no meio de um caminho de rede em que não tem negócios, de acordo com a pilha de protocolo. Neste caso, a frente fim é melhor considerado uma parte interna de um farm de servidores que encerra todas as camadas até a camada de aplicativo (e, portanto, pode usar todas as informações do cabeçalho para essas camadas).

No entanto, como com o NAT, esse design é útil na prática. A razão para olhar para os cabeçalhos TCP é que é possível fazer um trabalho melhor de balanceamento de carga do que apenas com informações de IP. Por exemplo, um endereço IP pode representar um en- empresa de pneus e fazer muitos pedidos. É apenas olhando para TCP ou superior informações da camada de que essas solicitações podem ser mapeadas para servidores diferentes. A razão para olhar os cabeçalhos HTTP é um pouco diferente. Muitos As interações na web acessam e atualizam bancos de dados, como quando um cliente consulta sua compra mais recente. O servidor que atender a esta solicitação terá que consultar o banco de dados back-end. É útil direcionar solicitações subsequentes do mesmo usuário para

Página 765

SEC. 7,5

ENTREGA DE CONTEÚDO

741

o mesmo servidor, porque esse servidor já armazenou informações sobre o do utilizador. A maneira mais simples de fazer com que isso aconteça é usar cookies da Web (ou outros informações para distinguir o usuário) e para inspecionar os cabeçalhos HTTP para encontrar o biscoitos.

Como uma nota final, embora tenhamos descrito este design para sites, um servidor O farm também pode ser criado para outros tipos de servidores. Um exemplo é o fluxo de servidores mídia sobre UDP. A única alteração necessária é que o front end seja capaz de balancear a carga dessas solicitações (que terão cabeçalhos de protocolo diferentes campos do que as solicitações da Web).

Web Proxies

As solicitações e respostas da Web são enviadas usando HTTP. Na seção 7.3, nós descrevemos como os navegadores podem armazenar respostas em cache e reutilizá-las para responder a solicitações futuras. Var- campos de cabeçalho ious e regras são usados pelo navegador para determinar se uma cópia em cache

de uma página da Web ainda está fresco. Não vamos repetir esse material aqui. O armazenamento em cache melhora o desempenho reduzindo o tempo de resposta e reduzindo a carga da rede. Se o navegador puder determinar que uma página em cache foi atualizada por ele-próprio, a página pode ser obtida do cache imediatamente, sem tráfego de rede em absoluto. No entanto, mesmo que o navegador deva solicitar ao servidor a confirmação de que a página ainda está fresca, o tempo de resposta é reduzido e a carga da rede é reduzida, especialmente para páginas grandes, uma vez que apenas uma pequena mensagem precisa ser enviada.

No entanto, o melhor que o navegador pode fazer é armazenar em cache todas as páginas da Web que o

o usuário já visitou. De nossa discussão sobre popularidade, você deve se lembrar que além de algumas páginas populares que muitas pessoas visitam repetidamente, há muitas, muitas páginas impopulares. Na prática, isso limita a eficácia do navegador cache porque há um grande número de páginas que são visitadas apenas uma vez por um determinado usuário. Essas páginas sempre devem ser buscadas no servidor.

Uma estratégia para tornar os caches mais eficazes é compartilhar o cache entre vários usuários múltiplos. Dessa forma, uma página já buscada por um usuário pode ser retornada para um outro usuário quando esse usuário faz a mesma solicitação. Sem o cache do navegador, ambos os usuários precisariam buscar a página no servidor. Claro, este compartilhamento não pode ser feito para tráfego criptografado, páginas que requerem autenticação e não armazenável em cache páginas (por exemplo, preços de ações atuais) que são retornadas por programas. Páginas dinâmicas criados por programas, especialmente, são um caso crescente para o qual o cache não é eficaz ffectivo. No entanto, existem muitas páginas da Web que são visíveis para muitos usuários e têm a mesma aparência, independentemente do usuário que faça a solicitação (por exemplo, imagens).

Um **proxy da Web** é usado para compartilhar um cache entre os usuários. Um proxy é um agente que atua em nome de outra pessoa, como o usuário. Existem muitos tipos de proxies. Por exemplo, um proxy ARP responde a solicitações ARP em nome de um usuário que é em outro lugar (e não pode responder por si mesmo). Um proxy da Web busca solicitações da Web em nome de seus usuários. Normalmente fornece armazenamento em cache das respostas da Web, e desde ele é compartilhado entre os usuários e tem um cache substancialmente maior do que um navegador.

Página 766

742

A CAMADA DE APLICAÇÃO INDIVÍDUO. 7

Quando um proxy é usado, a configuração típica é para uma organização operar um Proxy da Web para todos os seus usuários. A organização pode ser uma empresa ou um ISP. Ambos podem se beneficiar ao acelerar as solicitações da Web para seus usuários e reduzir seu necessidades de largura de banda. Embora o preço fixo, independente do uso, seja comum para usuários, a maioria das empresas e ISPs são cobrados de acordo com a largura de banda que eles usar.

Essa configuração é mostrada na Fig. 7-66. Para usar o proxy, cada navegador é configurado vermelho para fazer solicitações de página ao proxy em vez de ao servidor real da página. Se o proxy tem a página, ele retorna a página imediatamente. Se não, ele busca a página do servidor, adiciona-o ao cache para uso futuro e o devolve ao cliente que solicitou.

Clientes
Servidores
Cache do navegador
Organização
Cache proxy
Internet

Figura 7-66. Um cache de proxy entre navegadores da Web e servidores da Web.

Além de enviar solicitações da Web para o proxy em vez do servidor real, clients realizam seu próprio cache usando o cache do navegador. O proxy é apenas consultado depois que o navegador tentou satisfazer a solicitação de seu próprio cache. Isso é, o proxy fornece um segundo nível de armazenamento em cache.

Proxies adicionais podem ser adicionados para fornecer níveis adicionais de cache. Cada proxy (ou navegador) faz solicitações por meio de seu **proxy upstream**. Cada proxy upstream caches para os **proxies downstream** (ou navegadores). Assim, é possível para sers em uma empresa para usar um proxy de empresa, que usa um proxy de ISP, que tata servidores Web diretamente. No entanto, o único nível de cache de proxy que temos mostrado na Fig. 7-66 é frequentemente suficiente para obter a maioria dos benefícios potenciais, em prática. O problema novamente é a longa cauda da popularidade. Estudos de tráfego da web mostraram que o cache compartilhado é especialmente benéfico até que o número de usuários

atinge aproximadamente o tamanho de uma pequena empresa (digamos, 100 pessoas). Como o número de pessoas ficam maiores, os benefícios de compartilhar um cache tornam-se marginais devido ao os pedidos impopulares que não podem ser armazenados em cache devido à falta de espaço de armazenamento (Wolman et al., 1999).

Os proxies da Web fornecem benefícios adicionais que muitas vezes são um fator na decisão para implantá-los. Um benefício é filtrar o conteúdo. O administrador pode configurar

Página 767

SEC. 7,5

ENTREGA DE CONTEÚDO

743

o proxy para colocar sites na lista negra ou filtrar as solicitações que ele faz. Para exemplo, muitos administradores desaprovam os funcionários que assistem a vídeos do YouTube (ou pior ainda, pornografia) no horário comercial e definir seus filtros de acordo. A outro benefício de ter proxies é a privacidade ou anonimato, quando o proxy protege a identidade do usuário do servidor.

7.5.3 Redes de distribuição de conteúdo

Os farms de servidores e os proxies da Web ajudam a construir grandes sites e a melhorar desempenho, mas não são suficientes para sites realmente populares que devem servir conteúdo em escala global. Para esses sites, uma abordagem diferente é necessária.

CDNs (Content Delivery Networks) mudam a ideia do tradicional Web caching em sua cabeça. Em vez disso, os clientes procuram uma cópia da página solicitada em um cache próximo, é o provedor que coloca uma cópia da página em um conjunto de nós em locais diferentes e direciona o cliente a usar um nó próximo como servidor.

Um exemplo do caminho que os dados seguem quando são distribuídos por um CDN é mostrado na Fig. 7-67. É uma árvore. O servidor de origem no CDN distribui uma cópia do conteúdo para outros nós do CDN, em Sydney, Boston e Amsterdã, neste exemplo. Isso é mostrado com linhas tracejadas. Os clientes então buscam as páginas do nó mais próximo no CDN. Isso é mostrado com linhas sólidas. Desta forma, os clientes em Sydney, ambos buscam a cópia da página armazenada em Sydney; ambos não buscam a página do servidor de origem, que pode ser na Europa.

Origem CDN
servidor
Nó CDN
Sydney
Boston
Amsterdam
Distribuição para
Nós CDN
Página
buscar
Clientes mundiais

Figura 7-67. Árvore de distribuição do CDN.

Usar uma estrutura de árvore tem três virtudes. Primeiro, a distribuição de conteúdo pode ser ampliado para quantos clientes forem necessários, usando mais nós no CDN e mais níveis na árvore quando a distribuição entre os nós CDN torna-se o gargalo. Não importa quantos clientes existam, a estrutura em árvore é eficiente.

O servidor de origem não está sobrecarregado porque ele fala com muitos clientes através da árvore

Página 768

744

A CAMADA DE APLICAÇÃO
INDIVÍDUO. 7

de nós CDN; ele não precisa responder a cada solicitação de página por si só. Segundo, cada cliente obtém um bom desempenho buscando páginas de um servidor próximo em vez de um servidor distante. Isso ocorre porque o tempo de ida e volta para configurar um conexão é mais curta, o TCP de inicialização lenta aumenta mais rapidamente por causa do tempo de ida e volta, e o caminho de rede mais curto tem menos probabilidade de passar por regiões

de congestionamento na Internet. Finalmente, a carga total que é colocada na rede também é mantido no mínimo. Se os nós CDN estiverem bem posicionados, o tráfego para um determinada página deve passar por cada parte da rede apenas uma vez. Isso é importante porque alguém paga pela largura de banda da rede, eventualmente.

A ideia de usar uma árvore de distribuição é direta. O que é menos simples é como organizar os clientes para usar esta árvore. Por exemplo, servidores proxy seriam parecem fornecer uma solução. Olhando para a Figura 7-67, se cada cliente foi configurado para usar o nó CDN Sydney, Boston ou Amsterdam como um proxy da Web de cache, o a distribuição seguiria a árvore. No entanto, esta estratégia fica aquém na prática, por três razões. A primeira razão é que os clientes em uma determinada parte da rede provavelmente pertencem a diferentes organizações, então provavelmente estão usando diferentes Proxies da Web. Lembre-se de que os caches geralmente não são compartilhados entre as organizações antes de

causa do benefício limitado de armazenamento em cache em um grande número de clientes, e para se-

razões de curiosidade também. Em segundo lugar, pode haver vários CDNs, mas cada cliente usa apenas

um único cache de proxy. Qual CDN um cliente deve usar como seu proxy? Finalmente, portanto a questão mais prática de todas é que os proxies da Web são configurados pelos clientes. Eles podem ou não ser configurados para beneficiar a distribuição de conteúdo por um CDN, e há pouco que o CDN possa fazer a respeito.

Outra maneira simples de apoiar uma árvore de distribuição com um nível é usar **mirroring**. Nesta abordagem, o servidor de origem replica o conteúdo nos nós CDN como antes. Os nós CDN em diferentes regiões de rede são chamados de **espelhos**. As páginas da web no servidor de origem contêm links explícitos para os diferentes espelhos, geralmente informando ao usuário sua localização. Este design permite que o usuário selecione manualmente um espelho próximo para usar para baixar o conteúdo. Um uso típico de espelhamento é colocar um grande pacote de software em espelhos localizados, por exemplo, no Oriente e Costas oeste dos EUA, Ásia e Europa. Sites espelhados são geralmente completamente estático, e a escolha dos locais permanece estável por meses ou anos. Eles são uma técnica experimentada e testada. No entanto, eles dependem do usuário para fazer a distribuição porque os espelhos são sites realmente diferentes, mesmo que estejam interligados.

A terceira abordagem, que supera as dificuldades das duas anteriores, processa, usa DNS e é chamado de **redirecionamento de DNS**. Suponha que um cliente deseja para buscar uma página com o URL <http://www.cdn.com/page.html>. Para buscar a página, o navegador usará DNS para resolver www.cdn.com para um endereço IP. Este DNS a pesquisa prossegue da maneira usual. Usando o protocolo DNS, o navegador descobre o endereço IP do servidor de nomes para *cdn.com* e entra em contato com o nome servidor para solicitar que resolva www.cdn.com. Agora vem a parte realmente inteligente, o servidor de nomes é executado pelo CDN. Em vez de retornar o mesmo endereço IP para cada pedido, ele vai olhar para o endereço IP do cliente que fez o pedido e retornar

SEC. 7,5

ENTREGA DE CONTEÚDO

745

respostas diferentes. A resposta será o endereço IP do nó CDN que é mais próximo do cliente. Ou seja, se um cliente em Sydney pede ao servidor de nomes CDN para resolver www.cdn.com, o servidor de nomes retornará o endereço IP de Sydney Nó CDN, mas se um cliente em Amsterdã fizer a mesma solicitação, o servidor de nomes retornará o endereço IP do nó CDN de Amsterdã.

Essa estratégia é perfeitamente legal de acordo com a semântica do DNS. Nós temos visto anteriormente que os servidores de nomes podem retornar listas variáveis de endereços IP. Depois de

a resolução do nome, o cliente de Sydney buscará a página diretamente do Sydney CDN. Outras páginas no mesmo "servidor" serão obtidas diretamente de

o nó CDN de Sydney também por causa do cache DNS. A sequência geral de etapas são mostradas na Figura 7-68.

Origem CDN
servidor
2: Consultar DNS
DNS CDN
servidor
Amsterdam
Nó CDN
Sydney
Nó CDN
3: "Contate Sydney"
"Contact Amsterdam"
4: Buscar
página
1: Distribuir conteúdo
Clientes de Sydney
Clientes de Amsterdam

Figura 7-68. Direcionar clientes para nós CDN próximos usando DNS.

Uma questão complexa no processo acima é o que significa encontrar o mais próximo Nó CDN e como proceder. Para definir mais próximo, não é realmente geografia aquilo importa. Existem pelo menos dois fatores a serem considerados no mapeamento de um cliente para um

Nó CDN. Um fator é a distância da rede. O cliente deve ter um curto e caminho de rede de alta capacidade para o nó CDN. Esta situação produzirá rápido Transferências. Os CDNs usam um mapa que eles calcularam anteriormente para traduzir entre o endereço IP de um cliente e sua localização na rede. O nó CDN que é selecionado pode ser o que está na distância mais curta em linha reta ou não. o que importa é alguma combinação do comprimento do caminho da rede e qualquer capacidade limites ao longo dele. O segundo fator é a carga que já está sendo transportada pelo Nó CDN. Se os nós CDN estiverem sobrecarregados, eles fornecerão respostas lentas, assim como o servidor Web sobrecarregado que procuramos evitar em primeiro lugar. Assim, pode ser necessário equilibrar a carga entre os nós CDN, mapeando alguns clientes para nós que estão um pouco mais distantes, mas com carga mais leve. As técnicas de uso de DNS para distribuição de conteúdo foram iniciadas por Akamai a partir de 1998, quando a Web estava gemendo sob a carga de seus primeiros

Página 770

746

A CAMADA DE APLICAÇÃO

INDIVÍDUO. 7

crescimento. Akamai foi o primeiro grande CDN e se tornou o líder do setor. Proba-muito mais inteligente do que a ideia de usar DNS para conectar clientes a clientes próximos nodes era a estrutura de incentivos de seus negócios. As empresas pagam a Akamai para de-entregar seu conteúdo aos clientes, para que eles tenham sites responsivos que os clientes gostaria de usar. Os nós CDN devem ser colocados em locais de rede com boas condições conectividade, que inicialmente significava dentro de redes ISP. Para os ISPs, há uma vantagem apto a ter um nó CDN em suas redes, ou seja, que o nó CDN reduza a quantidade de largura de banda de rede upstream que eles precisam (e devem pagar), apenas como acontece com os caches de proxy. Além disso, o nó CDN melhora a capacidade de resposta para o

Os clientes do ISP, o que faz o ISP parecer bem aos seus olhos, dando-lhes um vantagem competitiva sobre ISPs que não têm um nó CDN. Esses benefícios (em nenhum custo para o ISP) torna a instalação de um nó CDN uma escolha óbvia para o ISP. Portanto, o provedor de conteúdo, o ISP e os clientes se beneficiam e o CDN torna dinheiro. Desde 1998, outras empresas entraram no negócio, então agora é um indústria competitiva com vários fornecedores.

Como esta descrição indica, a maioria das empresas não constrói seu próprio CDN. No-em vez disso, eles usam os serviços de um provedor de CDN, como a Akamai, para realmente fornecer seu conteúdo. Para permitir que outras empresas usem o serviço de um CDN, precisamos adicionar um último passo para nossa foto.

Depois que o contrato é assinado para um CDN distribuir conteúdo em nome de um Proprietário do site, o proprietário fornece ao CDN o conteúdo. Este conteúdo é enviado para os nós CDN. Além disso, o proprietário reescreve qualquer uma de suas páginas da Web com links para o conteúdo. Em vez de vincular ao conteúdo do site, as páginas vinculam a o conteúdo por meio do CDN. Como um exemplo de como esse esquema funciona, considere o código-fonte da página da Web do Fluffy Video, fornecido na Figura 7.69 (a). Após o pré-processamento, ele é transformado na Fig. 7-69 (b) e colocado no servidor do Fluffy Video como www.fluffyvideo.com/index.html.

Quando um usuário digita a URL www.fluffyvideo.com em seu navegador, o DNS re-transforma o endereço IP do próprio site do Fluffy Video, permitindo que o principal (HTML) página a ser buscada da maneira normal. Quando o usuário clica em qualquer um dos links, o navegador pede ao DNS para pesquisar www.cdn.com. Esta pesquisa entra em contato com o Servidor DNS do CDN, que retorna o endereço IP do nó CDN próximo. O navegador, em seguida, envia uma solicitação HTTP regular para o nó CDN, por exemplo, para fluffyvideo/koalas.mpg. O URL identifica a página a retornar, iniciando o caminho com *fluffyvideo* para que o nó CDN possa separar os pedidos para os diferentes componentes empresas a que serve. Finalmente, o vídeo é retornado e o usuário vê um fofo fofo animais.

A estratégia por trás dessa divisão de conteúdo hospedado pelo CDN e páginas de entrada hospedado pelo proprietário do conteúdo é que dá ao proprietário do conteúdo o controle enquanto permite o CDN mover a maior parte dos dados. A maioria das páginas de entrada são minúsculas, sendo apenas HTML texto. Essas páginas costumam ter links para arquivos grandes, como vídeos e imagens. É pré-especificamente esses arquivos grandes que são servidos pelo CDN, embora o uso de um CDN é completamente transparente para os usuários. O site parece o mesmo, mas executa mais rápido.

Página 771

SEC. 7,5

ENTREGA DE CONTEÚDO

747

```
<html>
<head> <title> Vídeo Fluffy </title> </head>
<body>
<h1> Lista de produtos do Fluffy Video </h1>
<p> Clique abaixo para obter amostras grátis. </p>
<a href="koalas.mpg"> Koalas hoje </a> <br>
<a href="kangaroos.mpg"> Cangurus engraçados </a> <br>
<a href="wombats.mpg"> Bonitos Wombats </a> <br>
</body>
</html>
(uma)
<html>
<head> <title> Vídeo Fluffy </title> </head>
<body>
<h1> Lista de produtos do Fluffy Video </h1>
<p> Clique abaixo para obter amostras grátis. </p>
<a href="http://www.cdn.com/fluffyvideo/koalas.mpg"> Koalas hoje </a> <br>
<a href="http://www.cdn.com/fluffyvideo/kangaroos.mpg"> Cangurus engraçados </a> <br>
<a href="http://www.cdn.com/fluffyvideo/wombats.mpg"> Nice Wombats </a> <br>
</body>
</html>
(b)
```

Figura 7-69. (a) Página da Web original. (b) Mesma página após vincular ao CDN.

Há outra vantagem para sites que usam um CDN compartilhado. A demanda futura para um site pode ser difícil de prever. Freqüentemente, há picos de demanda conhecido como **flash crowds**. Esse surto pode acontecer quando o produto mais recente é relocado, há desfile de moda ou outro evento, ou a empresa está no notícias. Até mesmo um site que era anteriormente desconhecido e não visitado pode de repente se tornar o foco da Internet se for interessante e tiver links de

sites populares. Uma vez que a maioria dos sites não está preparada para lidar com aumentos massivos em tráfego, o resultado é que muitos deles travam quando o tráfego aumenta. Caso em questão. Normalmente, o site do Secretário de Estado da Flórida não é um site ocupado local, embora você possa pesquisar informações sobre corporações da Flórida, notários, e assuntos culturais, bem como informações sobre votação e eleições lá. Para alguma razão estranha, em 7 de novembro de 2000 (a data da eleição presidencial dos EUA com Bush vs. Gore), um monte de gente de repente se interessou pela eleição página de resultados deste site. O site de repente se tornou um dos sites mais ocupados em o mundo e naturalmente caiu como resultado. Se estivesse usando um CDN, provavelmente sobreviveram. Ao usar um CDN, um site tem acesso a uma grande capacidade de serviço de conteúdo. Os maiores CDNs têm dezenas de milhares de servidores implantados em todos os países o mundo. Uma vez que apenas um pequeno número de sites terá uma multidão instantânea

Página 772

748

A CAMADA DE APLICAÇÃO

INDIVÍDUO. 7

a qualquer momento (por definição), esses sites podem usar a capacidade do CDN para lidar com a carga até a tempestade passar. Ou seja, o CDN pode escalar rapidamente um site capacidade de serviço.

A discussão anterior acima é uma descrição simplificada de como a Akamai trabalha. Existem muitos outros detalhes importantes na prática. A imagem de nós CDN turados em nosso exemplo são normalmente clusters de máquinas. O redirecionamento de DNS foi feito

com dois níveis: um para mapear clientes para a localização aproximada da rede, e um outro para distribuir a carga sobre os nós naquele local. Confiabilidade e desempenho mance são preocupações. Ser capaz de mudar um cliente de uma máquina em um cluster para outro, as respostas de DNS no segundo nível são fornecidas com TTLs curtos para que o client irá repetir a resolução após um curto período. Finalmente, embora tenhamos concentrados na distribuição de objetos estáticos como imagens e vídeos, os CDNs também podem oferecer suporte

criação de página dinâmica, mídia de streaming e muito mais. Para mais informações sobre CDNs, ver Dilley et al. (2002).

7.5.4 Redes Peer-to-Peer

Nem todos podem configurar um CDN de 1000 nós em locais ao redor do mundo para distribuir seu conteúdo. (Na verdade, não é difícil alugar 1000 máquinas virtuais em todo o mundo por causa da indústria de hospedagem bem desenvolvida e competitiva.

No entanto, a configuração de um CDN só começa com a obtenção dos nós.) Felizmente, há uma alternativa para o resto de nós que é simples de usar e pode distribuir um tremendo grande quantidade de conteúdo. É uma rede P2P (Peer-to-Peer).

As redes P2P entraram em cena a partir de 1999. A primeira aplicação generalizada plicação era para crimes em massa: 50 milhões de usuários do Napster trocavam cópias corrigiu músicas sem a permissão dos proprietários dos direitos autorais até que o Napster fosse fechado

pelos tribunais em meio a grande controvérsia. No entanto, a tecnologia peer-to-peer Gy tem muitos usos interessantes e legais. Outros sistemas continuaram o desenvolvimento, com tanto interesse dos usuários que o tráfego P2P rapidamente eclipsou o tráfego da web. Hoje, o BitTorrent é o protocolo P2P mais popular. É amplamente usado para compartilhar (licenciados e de domínio público) vídeos, bem como outros conteúdos, por ele responsáveis uma grande fração de todo o tráfego da Internet. Veremos isso nesta seção.

A ideia básica de uma rede de compartilhamento de arquivos **P2P (Peer-to-Peer)** é que muitos computadores se reúnem e reúnem seus recursos para formar uma distribuição de conteúdo sistema. Os computadores geralmente são simplesmente computadores domésticos. Eles não precisam ser

máquinas em centros de dados da Internet. Os computadores são chamados de pares porque cada

pode-se atuar alternadamente como um cliente para outro par, buscando seu conteúdo, e como um servidor, fornecendo conteúdo a outros pares. O que torna os sistemas ponto a ponto inter-O fato é que não há infraestrutura dedicada, ao contrário de um CDN. Todos participa da tarefa de distribuição de conteúdo, e muitas vezes não há um ponto central de ao controle.

Muitas pessoas estão entusiasmadas com a tecnologia P2P porque ela é vista como ering o rapaz. A razão não é apenas que é necessária uma grande empresa para administrar um

Página 773

SEC. 7,5

ENTREGA DE CONTEÚDO

749

CDN, enquanto qualquer pessoa com um computador pode se conectar a uma rede P2P. É aquela rede P2P

obras têm uma capacidade formidável de distribuir conteúdo que pode corresponder às maiores de sites.

Considere uma rede P2P composta por N usuários médios, cada um com banda larga conectividade a 1 Mbps. A capacidade de upload agregada da rede P2P, ou taxa em que os usuários podem enviar tráfego para a Internet, é N Mbps. A transferência capacidade, ou taxa na qual os usuários podem receber tráfego, também é N Mbps. Cada usuário podem fazer upload e download ao mesmo tempo, porque eles têm um link de 1 Mbps em cada direção.

Não é óbvio que isso deva ser verdade, mas acontece que toda a capacidade pode ser usado de forma produtiva para distribuir conteúdo, mesmo no caso de compartilhamento de gle cópia de um arquivo com todos os outros usuários. Para ver como isso pode ser, imagine que os usuários são organizados em uma árvore binária, com cada usuário não folha enviando para dois outros usuários. A árvore levará a única cópia do arquivo para todos os outros usuários. Para use a largura de banda de upload de tantos usuários quanto possível em todos os momentos (e, portanto, dis-

tributar o arquivo grande com baixa latência), precisamos canalizar a atividade de rede de os usuários. Imagine que o arquivo está dividido em 1000 partes. Cada usuário pode receber uma nova peça de algum lugar na árvore e envie a peça recebida anteriormente descer a árvore ao mesmo tempo. Dessa forma, uma vez que o pipeline é iniciado, após um pequeno número de peças (igual à profundidade da árvore) são enviadas, todos os usuários não-folha estará ocupado enviando o arquivo para outros usuários. Uma vez que existem aproximadamente N / 2

usuários não-folha, a largura de banda de upload desta árvore é N / 2 Mbps. Podemos repetir isso enganar e criar outra árvore que use os outros N / 2 Mbps de largura de banda de upload trocando as funções de nós folha e não folha. Juntos, esta construção usa todos da capacidade.

Esse argumento significa que as redes P2P são autoescaláveis. Seu upload utilizável

a capacidade cresce em conjunto com as demandas de download que podem ser feitas por seus Comercial. Eles são sempre "grandes o suficiente" em algum sentido, sem a necessidade de qualquer

infraestrutura dedicada. Em contraste, a capacidade de até mesmo um grande site é fixa e será muito grande ou muito pequeno. Considere um site com apenas 100 clusters, cada um capaz de 10 Gbps. Esta enorme capacidade não ajuda quando há um pequeno número de usuários. O site não consegue obter informações para N usuários em uma taxa mais rápida

do que N Mbps porque o limite está nos usuários e não no site. E quando há mais de um milhão de usuários de 1 Mbps, o site não pode bombear dados rápido o suficiente para manter todos os usuários ocupados baixando. Isso pode parecer um grande número de usuários, mas grandes redes BitTorrent (por exemplo, Pirate Bay) afirmam ter mais de 10.000.000 de usuários. Isso é mais como 10 terabits / s em termos de nosso ex-amplo!

Você deve pegar esses números no verso do envelope com um grão (ou melhor ainda, uma tonelada métrica) de sal porque eles simplificam a situação. Um significativo

desafio para redes P2P é usar bem a largura de banda quando os usuários podem entrar em todos formatos e tamanhos e têm diferentes capacidades de download e upload. Nunca menos, esses números indicam o enorme potencial do P2P.

Página 774

750

A CAMADA DE APLICAÇÃO INDIVÍDUO. 7

Há outra razão pela qual as redes P2P são importantes. CDNs e outros serviços gerenciados centralmente colocam os provedores em uma posição de ter um tesouro de informações sobre muitos usuários, de preferências de navegação e onde as pessoas compram online, para a localização das pessoas e endereços de e-mail. Esta informação pode ser usada para fornecer um serviço melhor e mais personalizado, ou pode ser usado para interferir nas pessoas privacidade. O último pode acontecer intencionalmente - digamos, como parte de um novo produto uct - ou através de uma divulgação acidental ou compromisso. Com sistemas P2P, há não pode haver um único provedor capaz de monitorar todo o sistema. Isto não significa que os sistemas P2P irão necessariamente fornecer privacidade, pois os usuários são confiando um no outro até certo ponto. Significa apenas que eles podem fornecer um diferente forma de privacidade do que sistemas gerenciados centralmente. Os sistemas P2P agora estão sendo explorado para serviços além do compartilhamento de arquivos (por exemplo, armazenamento, streaming) e tempo

diga se essa vantagem é significativa.

A tecnologia P2P seguiu dois caminhos relacionados à medida que foi desenvolvida. Em do lado mais prático, são os sistemas que são usados todos os dias. A maioria bem conhecidos desses sistemas são baseados no protocolo BitTorrent. No mais lado acadêmico, tem havido intenso interesse em DHT (Distributed Hash Table) algoritmos que permitem que os sistemas P2P funcionem bem como um todo, mas não dependem de nenhum cent

componentes racionalizados em tudo. Veremos essas duas tecnologias.

BitTorrent

O protocolo BitTorrent foi desenvolvido por Bram Cohen em 2001 para permitir que um conjunto de colegas compartilham arquivos com rapidez e facilidade. Existem dezenas de clients que falam este protocolo, assim como existem muitos navegadores que falam o HTTP protocolo para servidores web. O protocolo está disponível como um padrão aberto em www.bittorrent.org.

Em um sistema ponto a ponto típico, como aquele formado com o BitTorrent, os usuários cada um possui algumas informações que podem ser do interesse de outros usuários. Este informação pode ser software livre, música, vídeos, fotografias e assim por diante. tem três problemas que precisam ser resolvidos para compartilhar conteúdo nesta configuração:

1. Como um colega encontra outros pares que têm o conteúdo que deseja baixar?
2. Como o conteúdo é replicado por pares para fornecer downloads de alta velocidade para todos?
3. Como os colegas encorajam uns aos outros a fazer upload de conteúdo para outros como bem como baixar conteúdo para eles próprios?

O primeiro problema existe porque nem todos os pares terão todo o conteúdo, em pelo menos inicialmente. A abordagem adotada no BitTorrent é para cada provedor de conteúdo crie uma descrição de conteúdo chamada **torrent**. O torrent é muito menor que o

Página 775

SEC. 7,5 ENTREGA DE CONTEÚDO

751

conteúdo, e é usado por um par para verificar a integridade dos dados que ele baixa de outros pares. Outros usuários que desejam baixar o conteúdo devem primeiro obter o torrent, digamos, ao encontrá-lo em uma página da Web anunciando o conteúdo.

O torrent é apenas um arquivo em um formato específico que contém dois tipos de chave de em formação. Um tipo é o nome de um **rastreador**, que é um servidor que conduz pares para o conteúdo do torrent. O outro tipo de informação é uma lista de tamanhos iguais pedaços, ou **pedaços**, que compõem o conteúdo. Diferentes tamanhos de pedaços podem ser usados para

torrents diferentes, normalmente de 64 KB a 512 KB. O arquivo torrent contém o nome de cada pedaço, dado como um hash SHA-1 de 160 bits do pedaço. Vamos cobrir hashes criptográficos, como SHA-1 no cap. 8. Por enquanto, você pode pensar em um hash como uma soma de verificação mais longa e segura. Dado o tamanho dos pedaços e hashes, o arquivo torrent é pelo menos três ordens de magnitude menor que o conteúdo, então pode ser transferido rapidamente.

Para baixar o conteúdo descrito em um torrent, um par primeiro entra em contato com o rastreador para o torrent. O **rastreador** é um servidor que mantém uma lista de todos os outros pares que estão ativamente baixando e enviando o conteúdo. Este conjunto de pares é chamado de **enxame**. Os membros do enxame entram em contato com o rastreador regularmente para

relatam que ainda estão ativos, bem como quando saem do enxame. Quando um novo par contata o rastreador para se juntar ao enxame, o rastreador conta a ele sobre outros pares no enxame. Obter o torrent e entrar em contato com o rastreador são os dois primeiros etapas para baixar o conteúdo, conforme mostrado na Fig. 7-70.

Semente
par
Não chocado
pedaços
Tracker
Torrente
Par
1: Obter torrent
metarquivo
2: Obtenha pares
do rastreador
3: Partes comerciais
com colegas
Fonte de
conteúdo

Figura 7-70. BitTorrent.

O segundo problema é como compartilhar conteúdo de uma forma que permita um rápido download carrega. Quando um enxame é formado pela primeira vez, alguns pares devem ter todos os pedaços que

compõem o conteúdo. Esses pares são chamados de **semeadores**. Outros pares que se juntam ao enxame não terá pedaços; eles são os pares que estão baixando o conteúdo.

Enquanto um par participa de um enxame, ele simultaneamente baixa pedaços que está faltando em outros pares, e carrega pedaços que possui para outros pares que

752

A CAMADA DE APLICAÇÃO INDIVÍDUO. 7

preciso deles. Essa negociação é mostrada como a última etapa da distribuição de conteúdo na Fig. 7-

70. Com o tempo, o par reúne mais pedaços até que tenha baixado todos os conteúdo. O par pode deixar o enxame (e retornar) a qualquer momento. Normalmente um par permanecerá por um curto período após concluir seu próprio download. Com colegas chegando e indo, a taxa de rotatividade em um enxame pode ser bastante alta.

Para que o método acima funcione bem, cada pedaço deve estar disponível em muitos pares. Se todos obtivessem os pedaços na mesma ordem, é provável que muitos os pares dependeriam dos semeadores para a próxima parte. Isso criaria um gargalo. Em vez disso, os pares trocam listas dos blocos que possuem entre si.

Em seguida, eles selecionam pedaços raros que são difíceis de encontrar para baixar. A ideia é que baixar um pedaço raro fará uma cópia dele, o que tornará o pedaço fácil er para outros pares encontrarem e baixarem. Se todos os pares fizerem isso, depois de um curto tempo, todos

pedaços estarão amplamente disponíveis.

O terceiro problema é talvez o mais interessante. Nós CDN são configurados exclusivamente para fornecer conteúdo aos usuários. Nós P2P não. Eles são os usuários usuários, e os usuários podem estar mais interessados em obter um filme do que ajudar outros usuários com seus downloads. Nós que obtêm recursos de um sistema sem contribuir em espécie são chamados de **free-riders** ou **leechers**. Se houver muitos deles, o sistema não funcionará bem. Sistemas P2P anteriores eram conhecidos por hospedá-los (Saroiu et al., 2003) então o BitTorrent procurou minimizá-los.

A abordagem adotada em clientes BitTorrent é recompensar os pares que mostram bons comportamento de upload. Cada peer faz uma amostragem aleatória dos outros peers, recuperando pedaços

deles enquanto carrega pedaços para eles. O par continua a negociar pedaços com apenas um pequeno número de pares que fornecem o melhor desempenho de download, ao mesmo tempo que tenta aleatoriamente outros pares para encontrar bons parceiros. Tentando aleatoriamente peers também permite que os recém-chegados obtenham pedaços iniciais que podem negociar com outros pares. Os pares com os quais um nó está atualmente trocando blocos são considerados

não chocado.

Com o tempo, esse algoritmo tem como objetivo combinar pares com uploads semelhantes e taxas de download entre si. Quanto mais um colega está contribuindo para o outro colegas, mais ele pode esperar em retorno. Usar um conjunto de pares também ajuda a saturar um largura de banda de download do par para alto desempenho. Por outro lado, se um par não é enviar pedaços para outros pares, ou estiver fazendo isso muito lentamente, será interrompido ou **sufocado**, mais cedo ou mais tarde. Esta estratégia desencoraja o comportamento anti-social em que seus pares cavalgam livremente no enxame.

O algoritmo de sufocamento às vezes é descrito como a implementação do **tit-for-tat** estratégia que incentiva a cooperação em interações repetidas. No entanto, faz não impede que os clientes joguem com o sistema em qualquer sentido forte (Piatek et al., 2007). No entanto, atenção ao problema e aos mecanismos que o tornam mais difícil para usuários casuais viajarem de graça provavelmente contribuíram para o sucesso do Bit-Torrente.

Como você pode ver em nossa discussão, o BitTorrent vem com um vocabulário rico. Existem torrents, swarms, leechers, seeders e trackers, bem como snubbing,

SEC. 7,5

ENTREGA DE CONTEÚDO

753

asfixia, espreita e muito mais. Para obter mais informações, consulte o breve artigo sobre Bit-Torrent (Cohen, 2003) e procure na Web começando com www.bittorrent.org.

DHTs - Tabelas de Hash Distribuídas

O surgimento de redes de compartilhamento de arquivos P2P por volta de 2000 desencadeou muitas está na comunidade de pesquisa. A essência dos sistemas P2P é que eles evitam o estruturas gerenciadas centralmente de CDNs e outros sistemas. Isso pode ser significativo vantagem. Componentes gerenciados centralmente tornam-se um gargalo à medida que o sistema cresce muito e é um ponto único de falha. Os componentes centrais também podem ser usado como um ponto de controle (por exemplo, para desligar a rede P2P). No entanto, o início Os sistemas P2P eram apenas parcialmente descentralizados ou, se fossem totalmente descentralizados, eles eram ineficientes.

A forma tradicional de BitTorrent que acabamos de descrever usa ponto a ponto transferências e um rastreador centralizado para cada enxame. É o rastreador que resulta ser a parte difícil de descentralizar em um sistema ponto a ponto. O problema principal é como descobrir quais pares têm conteúdo específico que está sendo procurado. Para exemplo, cada usuário pode ter um ou mais itens de dados, como músicas, fotografias, programas, arquivos e assim por diante que outros usuários possam querer ler. Como fazer o outro

os usuários os encontram? Fazer um índice de quem tem o que é simples, mas é centralizado. Ter cada par manter seu próprio índice não ajuda. Verdade, é distribuído. No entanto, requer muito trabalho manter os índices de todos os pares até a data (conforme o conteúdo é movido pelo sistema) em que não vale a pena o esforço. A questão abordada pela comunidade de pesquisa foi se isso era possível para construir índices P2P totalmente distribuídos, mas com bom desempenho. Por portanto bem, queremos dizer três coisas. Primeiro, cada nó mantém apenas uma pequena quantidade de informações sobre outros nós. Esta propriedade significa que não será caro para manter o índice atualizado. Em segundo lugar, cada nó pode procurar entradas no índice rapidamente. Caso contrário, não é um índice muito útil. Terceiro, cada nó pode usar o índice ao mesmo tempo, mesmo quando outros nós vêm e vão. Esta propriedade significa que o desempenho do índice cresce com o número de nós.

A resposta para a pergunta era: "Sim". Quatro soluções diferentes foram ventilado em 2001. Eles são Chord (Stoica et al., 2001), CAN (Ratnasamy et al., 2001), Pastry (Rowstron e Druschel, 2001) e Tapestry (Zhao et al., 2004). Outras soluções foram inventadas logo depois, incluindo o Kademia, que é usado na prática (Maymounkov e Mazieres, 2002). As soluções são conhecidas como **DHTs** (**Tabelas de hash distribuídas**) porque a funcionalidade básica de um índice é mapear uma chave para um valor. Isso é como uma tabela hash, e as soluções são distribuídas na versão sões, é claro.

Os DHTs fazem seu trabalho impondo uma estrutura regular na comunicação entre os nós, como veremos. Este comportamento é bastante diferente daquele de redes P2P tradicionais que usam quaisquer conexões que os pares façam.

Página 778

754

A CAMADA DE APLICAÇÃO INDIVÍDUO. 7

Por esse motivo, os DHTs são chamados de **redes P2P estruturadas**. P2P tradicional - tocols constroem **redes P2P não estruturadas**.

A solução DHT que descreveremos é o Chord. Como cenário, considere como substituir o rastreador centralizado tradicionalmente usado no BitTorrent por um rastreador totalmente distribuído. O Chord pode ser usado para resolver esse problema. Nesse cenário, o índice geral é uma lista de todos os enxames que um computador pode juntar-se para baixar conteúdo. A chave usada para procurar o índice é a descrição do torrent do conteúdo. Ele identifica exclusivamente um enxame do qual o conteúdo pode ser baixado como hashes de todos os blocos de conteúdo. O valor armazenado no índice para cada chave está a lista de pares que compõem o enxame. Esses pares são os computadores para entrar em contato para baixar o conteúdo. Uma pessoa que deseja baixar conteúdo

como um filme, tem apenas a descrição do torrent. A pergunta que o DHT deve responder é como, sem um banco de dados central, uma pessoa descobre quais pares (fora dos milhões de nós do BitTorrent) para baixar o filme?

Um Chord DHT consiste em n nós participantes. Eles são nós executando o BitTorrent em nosso cenário. Cada nó tem um endereço IP pelo qual pode ser contactado. O índice geral é espalhado pelos nós. Isso implica que cada nó armazena bits e partes do índice para uso por outros nós. A parte chave do Chord é que ele navega no índice usando identificadores em um espaço virtual, não o anúncio de IP vestidos de nós ou nomes de conteúdo, como filmes. Conceitualmente, a identidade fílers são simplesmente números de m bits que podem ser organizados em ordem crescente em um anel.

Para transformar um endereço de nó em um identificador, ele é mapeado para um número m -bit usando uma função *hash*, *hash*. O Chord usa SHA-1 para *hash*. Este é o mesmo hash que mencionamos ao descrever o BitTorrent. Vamos olhar para isso quando dissermos sobre criptografia no cap. 8. Por enquanto, basta dizer que é apenas uma função que leva uma string de bytes de comprimento variável como um argumento e produz um

número dom 160 bits. Assim, podemos usá-lo para converter qualquer endereço IP em um endereço de 160 bits

número denominado **identificador de nó**.

Na Fig. 7-71 (a), mostramos o círculo identificador de nó para $m = 5$. (Basta ignorar os arcos no meio no momento.) Alguns dos identificadores correspondem a nós, mas a maioria não. Neste exemplo, os nós com identificadores 1, 4, 7, 12, 15, 20 e 27 correspondem a nós reais e estão sombreados na figura; o resto não existe.

Vamos agora definir o *sucessor* da função (k) como o identificador de nó do primeiro nó real seguindo k ao redor do círculo, no sentido horário. Por exemplo, *sucessor* (6) = 7 , *sucessor* (8) = 12 e *sucessor* (22) = 27 .

Uma **chave** também é produzida por hash de um nome de conteúdo com *hash* (ou seja, SHA-1) para gerar um número de 160 bits. Em nosso cenário, o nome do conteúdo é o torrent. Portanto, a fim de converter *torrent* (o arquivo de descrição do torrent) em sua chave, calculamos *chave* = *hash* (*torrent*). Este cálculo é apenas uma chamada de procedimento local para *hash*. Para iniciar um novo enxame, um nó precisa inserir um novo par de valores-chave consistindo de (*torrent*, *meu-endereço IP*) no índice. Para fazer isso, o nó pede *sucessor* (*hash* (*torrent*)) para armazenar *meu endereço IP*. Desta forma, o índice é distribuído sobre os nós aleatoriamente. Para tolerância a falhas, p funções hash diferentes

Página 779

SEC. 7,5

ENTREGA DE CONTEÚDO

755

0
16
2,3
4
4
4
3
2
2,3
0,1
0,1
0,1
31
1
17
15
2
30
14
18
29
3
19
13
4
28
12
20
27
26
25
24
8
7
23
9
6
22
10
5
21
11
2
4
3
4
Nó 1's
dedo
tabela
5
7

```

9
12
17
20
IP inicial
addr
do sucessor
5
7
6
7
Nó 4's
dedo
tabela
8
12
12
12
20
20
IP inicial
addr
do sucessor
13
15
14
15
Nodo 12
dedo
tabela
16
20
20
20
28
1
IP inicial
addr
do sucessor
(b)
(uma)
Nó
identificador
Real
nó

```

Figura 7-71. (a) Um conjunto de 32 identificadores de nós dispostos em um círculo. O sombreado aqueles correspondem a máquinas reais. Os arcos mostram os dedos dos nós 1, 4, e 12. As etiquetas nos arcos são os índices da tabela. (b) Exemplos do dedo tabelas.

poderia ser usado para armazenar os dados em p nós, mas não vamos considerar o assunto de tolerância a falhas mais aqui.

Algum tempo depois que o DHT é construído, outro nó quer encontrar um torrent para que ele possa se juntar ao enxame e baixar conteúdo. Um nó procura *torrent* por primeiro hashing para obter a *chave* e, segundo, usando o *sucessor* (*chave*) para encontrar o endereço IP

do nó armazenando o valor correspondente. O valor é a lista de pares no enxame; o nó pode adicionar seu endereço IP à lista e entrar em contato com os outros pares para baixar conteúdo com o protocolo BitTorrent.

A primeira etapa é fácil; o segundo não é fácil. Para tornar possível encontrar o endereço IP do nó correspondente a uma determinada chave, cada nó é necessário para

Página 780

756

A CAMADA DE APLICAÇÃO INDIVÍDUO. 7

manter certas estruturas de dados administrativos. Um deles é o endereço IP de seu nó sucessor ao longo do círculo identificador de nó. Por exemplo, na Figura 7-71, o sucessor do nó 4 é 7 e o sucessor do nó 7 é 12.

A pesquisa agora pode proceder da seguinte maneira. O nó solicitante envia um pacote para seu sucessor contém seu endereço IP e a chave que está procurando. O pacote é propagado ao redor do anel até localizar o sucessor para o identificador de nó sendo solicitado. Esse nó verifica se há alguma informação que corresponda à chave, e em caso afirmativo, retorna-o diretamente para o nó solicitante, cujo endereço IP ele possui.

No entanto, pesquisar linearmente todos os nós é muito ineficiente em um grande par sistema ponto a ponto, uma vez que o número médio de nós necessários por pesquisa é $n/2$. Para acelerar muito a pesquisa, cada nó também mantém o que Chord chama de **dedo mesa**. A tabela de dedo tem m entradas, indexadas por 0 a $m - 1$, cada um ponto-para um nó real diferente. Cada uma das entradas possui dois campos: *início* e o IP endereço do *sucessor* (*início*), conforme mostrado para três exemplos de nós na Figura 7.71 (b). Os valores dos campos para a entrada i em um nó com identificador k são:

$$\text{início} = k + 2^i \pmod{2^m}$$

Endereço IP do *sucessor* (*start* [i])

Observe que cada nó armazena os endereços IP de um número relativamente pequeno de nós e que a maioria deles está bem próxima em termos de identificador de nó.

Usando a tabela de dedos, a pesquisa da *chave* no nó k procede da seguinte forma. Se *chave* fica entre k e o *sucessor* (k), o nó que contém informações sobre a *chave* é *sucessor* (k) e a pesquisa termina. Caso contrário, a tabela de dedos é pesquisada para encontrar a entrada cujo campo *inicial* é o predecessor mais próximo da *chave*. Um pedido é então enviado diretamente para o endereço IP dessa entrada da tabela de dedo para solicitar que continue o pesquisa. Como está mais perto da *chave*, mas ainda abaixo dela, há boas chances de que seja capaz de retornar a resposta com apenas um pequeno número de consultas adicionais. De fato, uma vez que cada pesquisa reduz pela metade a distância restante até o alvo, ela pode ser mostrada que o número médio de pesquisas é $\log_2 n$.

Como primeiro exemplo, considere procurar a *chave* = 3 no nó 1. Como o nó 1 sabe que 3 está entre ele e seu sucessor, 4, o nó desejado é 4 e o a pesquisa termina, retornando o endereço IP do nó 4.

Como um segundo exemplo, considere procurar a *chave* = 16 no nó 1. Uma vez que 16 não se encontra entre 1 e 4, a tabela de dedo é consultada. O predecessor mais próximo de 16 é 9, então a solicitação é encaminhada para o endereço IP da entrada do 9, ou seja, o de nó 12. O nó 12 também não sabe a resposta em si, então procura o nó mais proximamente precedendo 16 e encontra 14, que produz o endereço IP do nó 15.

Uma consulta é enviada para lá. O nó 15 observa que 16 fica entre ele e seu cessor (20), então ele retorna o endereço IP 20 para o chamador, que funciona do seu jeito de volta ao nó 1.

Uma vez que os nós entram e saem o tempo todo, o Chord precisa de uma maneira de lidar com essas operações. Assumimos que quando o sistema começou a funcionar era pequeno o suficiente que os nós poderiam apenas trocar informações diretamente para construir o primeiro círculo e

Página 781

SEC. 7,5

ENTREGA DE CONTEÚDO

757

tabelas de dedo. Depois disso, é necessário um procedimento automatizado. Quando um novo nó, r , deseja entrar, ele deve entrar em contato com algum nó existente e solicitar que ele procure o IP endereço do *sucessor* (r) para ele. Em seguida, o novo nó pede ao *sucessor* (r) seu antecessor. O novo nó então pede a ambos para inserir r entre eles em o círculo. Por exemplo, se 24 na Figura 7.71 deseja se juntar, ele pede a qualquer nó para olhar *sucessor* ascendente (24), que é 27. Em seguida, ele pede 27 para seu predecessor (20). Depois disso conta a ambos sobre sua existência, 20 usa 24 como seu sucessor e 27 usa 24 como seu antecessor. Além disso, o nó 27 entrega essas chaves no intervalo 21-24, que agora pertencem a 24. Neste ponto, 24 está totalmente inserido.

No entanto, muitas tabelas de dedos estão erradas. Para corrigi-los, cada nó executa um processo em segundo plano que recalcula periodicamente cada dedo chamando *sucessor*. Quando uma dessas consultas atinge um novo nó, o dedo correspondente a entrada é atualizada.

Quando um nó sai normalmente, ele entrega suas chaves ao seu sucessor e informa seu antecessor de sua partida para que o predecessor possa se conectar ao departamento sucessor do nó ing. Quando um nó trava, surge um problema porque seu predecessor não tem mais um sucessor válido. Para aliviar este problema, cada nó mantém

acompanhar não só de seu sucessor direto, mas também seus sucessores diretos, para permitir que ele pule até $s - 1$ nó consecutivo com falha e reconecte o círculo em caso de desastre greves.

Tem havido uma enorme quantidade de pesquisas sobre DHTs desde que ventilado. Para lhe dar uma ideia de quanta pesquisa, deixe-nos fazer uma pergunta: qual é o artigo de rede mais citado de todos os tempos? Você achará difícil apresentar um artigo que seja mais citado do que o artigo seminal do Chord (Stoica et al., 2001). Apesar desta verdadeira montanha de pesquisas, as aplicações de DHTs são apenas lentamente começando a emergir. Alguns clientes BitTorrent usam DHTs para fornecer um rastreador totalmente distribuído do tipo que descrevemos. Grande nuvem comercial serviços como o Dynamo da Amazon também incorporam técnicas de DHT (DeCandia et al., 2007).

7,6 RESUMO

A nomenclatura na ARPANET começou de uma maneira muito simples: um arquivo de texto ASCII listou os nomes de todos os hosts e seus endereços IP correspondentes. Toda noite todas as máquinas baixaram este arquivo. Mas quando a ARPANET se transformou no Internet e explodiu em tamanho, uma nomenclatura muito mais sofisticada e dinâmica esquema era necessário. O usado agora é um esquema hierárquico chamado Do-Sistema de nomes principal. Ele organiza todas as máquinas da Internet em um conjunto de árvores. No nível superior estão os domínios genéricos bem conhecidos, incluindo *com* e *edu*, bem como cerca de 200 domínios de países. DNS é implementado como um banco de dados com servidores em todo o mundo. Ao consultar um servidor DNS, um processo

Página 782

758

A CAMADA DE APLICAÇÃO
INDIVÍDUO. 7

pode mapear um nome de domínio da Internet no endereço IP usado para se comunicar com um computador para esse domínio.

Email é o aplicativo matador original da Internet. Ainda é amplamente utilizado por todos um de crianças pequenas a avós. A maioria dos sistemas de e-mail do mundo usa o sistema de correio agora definido nas RFCs 5321 e 5322. As mensagens têm ASCII simples cabeçalhos e muitos tipos de conteúdo podem ser enviados usando MIME. O correio é enviado para agentes de transferência de mensagem para entrega e recuperada deles para apresentação por um uma variedade de agentes de usuário, incluindo aplicativos da Web. O correio enviado é entregue usando SMTP, que funciona fazendo uma conexão TCP a partir da mensagem de envio agente de transferência do sábio para o receptor.

A Web é o aplicativo que a maioria das pessoas pensa ser a Internet. Originalmente, era um sistema para vincular perfeitamente páginas de hipertexto (escritas em HTML) entre máquinas. As páginas são baixadas fazendo um TCP connection do navegador para um servidor e usando HTTP. Hoje em dia, muito do o conteúdo da Web é produzido dinamicamente, seja no servidor (por exemplo, com PHP) ou no navegador (por exemplo, com JavaScript). Quando combinado com dados de back-end-bases, páginas de servidor dinâmicas permitem aplicativos da Web, como e-commerce e pesquisa. As páginas dinâmicas do navegador estão evoluindo para aplicativos completos, como como e-mail, que roda dentro do navegador e usa os protocolos da Web para se comunicar com servidores remotos.

Cache e conexões persistentes são amplamente usados para melhorar o desempenho da Web performance. Usar a Web em dispositivos móveis pode ser desafiador, apesar do crescimento na largura de banda e poder de processamento dos celulares. Os sites costumam enviar versões personalizadas de páginas com imagens menores e navegação menos complexa para de-vícios com pequenos monitores.

Os protocolos da Web estão cada vez mais sendo usados para comunicações máquina a máquina comunicação. XML é preferível a HTML como uma descrição de conteúdo que é fácil para máquinas para processar. SOAP é um mecanismo RPC que envia mensagens XML usando HTTP.

Áudio e vídeo digital têm sido os principais impulsionadores da Internet desde 2000. O a maioria do tráfego da Internet hoje é de vídeo. Muito disso é transmitido de sites da Web sobre uma combinação de protocolos (incluindo RTP / UDP e RTP / HTTP / TCP). Mídia ao vivo é transmitido a muitos consumidores. Inclui estações de rádio e TV na Internet que transmitir todos os tipos de eventos. Áudio e vídeo também são usados para controle em tempo real referenciamento. Muitas chamadas usam voz sobre IP, em vez da rede telefônica tradicional trabalhar e incluir videoconferência.

Há um pequeno número de sites extremamente populares, bem como um número muito grande de menos populares. Para servir aos sites populares, distribuição de conteúdo redes de ação foram implantadas. Os CDNs usam DNS para direcionar clientes para um local próximo servidor; os servidores são colocados em centros de dados em todo o mundo. Alternativamente, As redes P2P permitem que uma coleção de máquinas compartilhe conteúdo, como filmes entre si mesmos. Eles fornecem uma capacidade de distribuição de conteúdo que escala com o número de máquinas na rede P2P e que pode rivalizar com o maior dos sites.

Página 783

INDIVÍDUO. 7
PROBLEMAS

759

PROBLEMAS

1. Muitos computadores comerciais têm três identificadores distintos e exclusivos em todo o mundo. o que são eles?
2. Na Fig. 7-4, não há período após o *laserjet* . Por que não?
3. Considere uma situação em que um ciberterrorista fabrica todos os servidores DNS do mundo travar simultaneamente. Como isso muda a capacidade de usar a Internet?
4. O DNS usa UDP em vez de TCP. Se um pacote DNS for perdido, não haverá recuperação automática. Isso causa um problema e, em caso afirmativo, como é resolvido?
5. John quer ter um nome de domínio original e usa um programa aleatório para gerar comeu um nome de domínio secundário para ele. Ele deseja registrar este nome de domínio no *com* domínio genérico. O nome de domínio gerado tem 253 caracteres. O registrador *com* permitirá que este nome de domínio seja registrado?
6. Uma máquina com um único nome DNS pode ter vários endereços IP? Como poderia isso ocorrer?
7. O número de empresas com um site na Web cresceu explosivamente nos últimos anos. Como resultado, milhares de empresas estão registradas no domínio *com* , causando um forte carregar no servidor de nível superior para este domínio. Sugira uma maneira de aliviar este problema sem alterar o esquema de nomenclatura (ou seja, sem introduzir um novo domínio de nível superior nomes). É permitido que sua solução exija alterações no código do cliente.
8. Alguns sistemas de e-mail oferecem suporte para *Retorno de conteúdo*: campo de cabeçalho. Ele especifica se o corpo de uma mensagem deve ser devolvido em caso de não entrega. Este campo pertence para o envelope ou para o cabeçalho?
9. Os sistemas de correio eletrônico precisam de diretórios para que os endereços de e-mail das pessoas possam ser consultados acima. Para construir esses diretórios, os nomes devem ser divididos em componentes padrão (por exemplo, nome, sobrenome) para tornar a pesquisa possível. Discuta alguns problemas que deve ser resolvido para que um padrão mundial seja aceitável.
10. Um grande escritório de advocacia, que tem muitos funcionários, fornece um único endereço de e-mail para cada empregado. O endereço de e-mail de cada funcionário é <*login*> @ *lawfirm.com* . No entanto, a empresa não definiu explicitamente o formato do login. Assim, alguns funcionários usam seus primeiros nomes como seus nomes de login, alguns usam seus sobrenomes, alguns usam suas iniciais, etc. A empresa agora deseja fazer um formato fixo, por exemplo:
firstname.lastname@lawfirm.com ,
que pode ser usado para os endereços de e-mail de todos os seus funcionários. Como isso pode ser feito sem balançar muito o barco?
11. Um arquivo binário tem 4560 bytes. Quanto tempo levará se codificado usando en- base64 codificação, com um par CR + LF inserido após cada 110 bytes enviados e no final?
12. Cite cinco tipos de MIME não listados neste livro. Você pode verificar seu navegador ou o Internet para obter informações.

Página 784

760

PROBLEMAS

INDIVÍDUO. 7

13. Suponha que você queira enviar um arquivo MP3 a um amigo, mas o ISP do seu amigo limita o tamanho de cada mensagem recebida para 1 MB e o arquivo MP3 é de 4 MB. Existe uma maneira de lidar com essa situação usando RFC 5322 e MIME?
14. Suponha que John acabou de configurar um mecanismo de encaminhamento automático em seu anúncio de e-mail comercial vestido, que recebe todos os seus e-mails relacionados a negócios, para encaminhá-los para sua pessoa-al endereço de e-mail, que ele compartilha com sua esposa. A esposa de John não sabia disso, e ativou um agente de férias em sua conta pessoal. Porque John encaminhou seu e-mail, ele não configurou um daemon de férias em sua máquina de trabalho. O que acontece quando um e-mail foi recebido no endereço de e-mail comercial de John?
15. Em qualquer padrão, como RFC 5322, uma gramática precisa do que é permitido é necessária para que diferentes implementações podem interagir. Mesmo itens simples devem ser definidos cuidadosamente. Os cabeçalhos SMTP permitem um espaço em branco entre os tokens. Dê *dois* plausíveis definições alternativas de espaço em branco entre tokens.
16. O agente de férias faz parte do agente do usuário ou do agente de transferência de mensagens? Claro, ele é configurado usando o agente do usuário, mas o agente do usuário realmente envia as respostas? Exclaramente sua resposta.
17. Em uma versão simples do algoritmo Chord para pesquisa ponto a ponto, as pesquisas não use a mesa de dedo. Em vez disso, eles são lineares ao redor do círculo, em qualquer direção. Pode um nó prediz com precisão em que direção ele deve pesquisar? Discuta sua resposta.
18. O IMAP permite que os usuários busquem e baixem e-mail de uma caixa de correio remota. Faz isso significar que o formato interno das caixas de correio deve ser padronizado para que qualquer protocolo IMAP grama no lado do cliente pode ler a caixa de correio em qualquer servidor de email? Discuta o seu resposta.
19. Considere o círculo Chord da Fig. 7-71. Suponha que o nó 18 fique online repentinamente. Quais das tabelas de dedos mostradas na figura são afetadas? como?
20. O Webmail usa POP3, IMAP ou nenhum dos dois? Se um desses, por que aquele escolheu sen? Se nenhum, de qual está mais próximo em espírito?
21. Quando as páginas da Web são enviadas, elas são prefixadas por cabeçalhos MIME. Por quê?
22. É possível que, quando um usuário clica em um link com o Firefox, um ajudante específico é iniciado-ed, mas clicar no mesmo link no Internet Explorer causa um efeito completamente diferente auxiliar a ser iniciado, embora o tipo MIME retornado em ambos os casos seja idêntico? Explique sua resposta.
23. Embora não tenha sido mencionado no texto, uma forma alternativa para um URL é usar o Endereço IP em vez de seu nome DNS. Use essas informações para explicar por que um nome DNS não pode terminar com um dígito.
24. Imagine que alguém do departamento de matemática de Stanford acabou de escrever um novo documento m, incluindo uma prova de que deseja distribuir por FTP para seus colegas revisarem. Ele coloca o programa no diretório FTP `ftp / pub / forReview / newProof.pdf`. O que é o URL deste programa provavelmente será?
25. Na Figura 7-22, www.aportal.com rastreia as preferências do usuário em um cookie. Uma desgraça A vantagem deste esquema é que os cookies são limitados a 4 KB, portanto, se as preferências forem

Página 785

INDIVÍDUO. 7

PROBLEMAS

761

extensa, por exemplo, muitas ações, equipes esportivas, tipos de notícias, previsão do tempo para várias cidades, promoções em várias categorias de produtos e muito mais, o limite de 4 KB pode ser alcançado. Projete uma maneira alternativa de controlar as preferências que não tem este problema.

26. O Sloth Bank deseja tornar o serviço bancário online fácil para seus clientes preguiçosos, então, após um mer se inscreve e é autenticado por uma senha, o banco retorna um cookie contendo um número de identificação do cliente. Desta forma, o cliente não precisa se identificar ou digite uma senha em futuras visitas ao banco online. O que você pensa dessa ideia? será que vai dar certo? É uma boa ideia?

27. (a) Considere a seguinte tag HTML:
`<h1 title = "este é o cabeçalho "> CABEÇALHO 1 </h1>`
Em que condições o navegador usa o atributo `TITLE` e como?
(b) Como o atributo `TITLE` difere do atributo `ALT`?

28. Como você torna uma imagem clicável em HTML? Dê um exemplo.
29. Escreva uma página HTML que inclua um link para o endereço de e-mail `nome de usuário @ Do-`

mainName.com . O que acontece quando um usuário clica neste link?

30. Escreva uma página XML para um registrador de universidade listando vários alunos, cada um tendo um nome, um endereço e um GPA.

31. Para cada uma das seguintes aplicações, diga se seria (1) possível e (2) melhor usar um script PHP ou JavaScript e por quê:

(a) Exibição de um calendário para qualquer mês solicitado desde setembro de 1752.

(b) Exibir a programação dos voos de Amsterdã para Nova York.

(c) Representar graficamente um polinômio a partir de coeficientes fornecidos pelo usuário.

32. Escreva um programa em JavaScript que aceite um número inteiro maior que 2 e diga se é um número primo. Observe que JavaScript tem instruções if e while com o mesmo sintaxe como C e Java. O operador do módulo $\% \text{.}$ Se você precisa da raiz quadrada de x , use $\text{Math.sqrt}(x)$.

33. Uma página HTML é a seguinte:

```
<html><body>
<a href="www.info-source.com/welcome.html"> Clique aqui para obter informações </a>
</body></html>
```

Se o usuário clicar no hiperlink, uma conexão TCP é aberta e uma série de linhas é enviada para o servidor. Liste todas as linhas enviadas.

34. O cabeçalho *If-Modified-Since* pode ser usado para verificar se uma página em cache ainda está válido. As solicitações podem ser feitas para páginas contendo imagens, som, vídeo e assim por diante, como bem como HTML. Você acha que a eficácia desta técnica é melhor ou pior para Imagens JPEG em comparação com HTML? Pense cuidadosamente sobre o que "eficácia" significa e explique sua resposta.

35. No dia de um grande evento esportivo, como o jogo do campeonato em algum esporte, muitas pessoas vão ao site oficial. Esta é uma multidão instantânea no mesmo sentido como a eleição presidencial de 2000 na Flórida? Por que ou por que não?

Página 786

762

PROBLEMAS

INDIVÍDUO. 7

36. Faz sentido que um único ISP funcione como um CDN? Se sim, como isso trabalhos? Se não, o que há de errado com a ideia?

37. Suponha que a compressão não seja usada para CDs de áudio. Quantos MB de dados devem ser contém um disco compacto para poder reproduzir duas horas de música?

38. Na Fig. 7-42 (c), o ruído de quantização ocorre devido ao uso de amostras de 4 bits para representar nove valores de sinal. A primeira amostra, em 0, é exata, mas as próximas não são. O que é o erro percentual para as amostras em 1/32, 2/32 e 3/32 do período?

39. Um modelo psicoacústico poderia ser usado para reduzir a largura de banda necessária para a Internet telefonia? Em caso afirmativo, quais condições, se houver, teriam de ser atendidas para que funcionasse? Se não, Por que não?

40. Um servidor de streaming de áudio tem uma "distância" unilateral de 100 ms para um reprodutor de mídia. Ele produz a 1 Mbps. Se o media player tiver um buffer de 2 MB, o que você pode dizer sobre a posição da marca d'água baixa e da marca d'água alta?

41. A voz sobre IP tem os mesmos problemas com firewalls que o streaming de áudio tem?
Discuta sua resposta.

42. Qual é a taxa de bits para a transmissão de quadros de cores não compactados de 1200×800 pixels com 16 bits / pixel a 50 quadros / s?

43. Um erro de 1 bit em um quadro MPEG pode afetar mais do que o quadro no qual o erro ocorreu maldito Explique sua resposta.

44. Considere um servidor de vídeo de 50.000 clientes, onde cada cliente assiste a três filmes por mês. Dois terços dos filmes são exibidos às 21h . M . Quantos filmes o servidor tem que transmitir de uma vez durante este período de tempo? Se cada filme exigir 6 Mbps, de quantas conexões OC-12 o servidor precisa para a rede?

45. Suponha que a lei de Zipf seja válida para acessos a um servidor de vídeo com 10.000 filmes. Se o servidor mantém os 1000 filmes mais populares na memória e os 9000 restantes no disco, forneça uma expressão para a fração de todas as referências que serão para a memória. Escreva um pequeno programa para avaliar esta expressão numericamente.

46. Alguns cybersquatters registraram nomes de domínio que são erros ortográficos comuns sites corporativos, por exemplo, www.microsfot.com. Faça uma lista de pelo menos cinco dessas rede.

47. Várias pessoas registraram nomes DNS que consistem em www.word.com , onde palavra é uma palavra comum. Para cada uma das seguintes categorias, liste cinco desses sites e resumir brevemente o que é (por exemplo, www.stomach.com pertence a um gastroenterologista em Long Island). Aqui está a lista de categorias: animais, alimentos, objetos domésticos, e partes do corpo. Para a última categoria, prefira as partes do corpo acima da cintura.

48. Reescreva o servidor da Fig. 6-6 como um verdadeiro servidor Web usando o comando *GET* para HTTP 1.1. Ele também deve aceitar a mensagem do *Host*. O servidor deve manter um cache de arquivos recentemente obtidos do disco e atendem às solicitações do cache, quando possível.

Página 787

8

SEGURANÇA DE REDE

Nas primeiras décadas de sua existência, as redes de computadores foram principalmente usado por pesquisadores universitários para enviar e-mail e por funcionários corporativos para compartilhamento de impressoras. Nessas condições, a segurança não recebeu muita atenção. Mas agora, como milhões de cidadãos comuns estão usando redes para bancos, lojas ping, e preencher suas declarações fiscais, e fraqueza após a fraqueza ter sido encontrada, a segurança da rede se tornou um problema de proporções gigantescas. Neste capítulo, vamos estudar a segurança da rede de vários ângulos, apontar inúmeras armadilhas, e discutir muitos algoritmos e protocolos para tornar as redes mais seguras.

Segurança é um tópico amplo e cobre uma infinidade de pecados. Em sua forma mais simples, preocupa-se em garantir que as pessoas intrometidas não consigam ler, ou pior ainda, modificar secretamente mensagens destinadas a outros destinatários. Está preocupado com as pessoas

ple tentando acessar serviços remotos que não estão autorizados a usar. Isso também trata de maneiras de saber se essa mensagem supostamente do IRS " Pague por Sexta-feira, senão " é realmente do IRS e não da Máfia. Segurança também lida com os problemas de mensagens legítimas sendo capturadas e reproduzidas, e com pessoas mais tarde tentando negar que eles enviaram certas mensagens.

A maioria dos problemas de segurança são causados intencionalmente por pessoas mal-intencionadas que tentam obter algum benefício, chamar a atenção ou prejudicar alguém. Alguns dos mais comuns os perpetradores estão listados na Fig. 8-1. Deve ficar claro a partir desta lista que fazer um rede segura envolve muito mais do que apenas mantê-la livre de programação rors. Envolve ser mais esperto que muitas vezes inteligente, dedicado e, às vezes, bem adversários financiados. Também deve ficar claro que as medidas que irão impedir

763

Página 788

764

SEGURANÇA DE REDE

INDIVÍDUO. 8

os invasores terão pouco impacto sobre os graves. Registros policiais mostram que o os ataques mais prejudiciais não são perpetrados por estranhos grampeando uma linha telefônica, mas

por insiders que guardam rancor. Os sistemas de segurança devem ser projetados de acordo.

Adversário

Objetivo

Aluna

Para se divertir bisbilhotando o e-mail das pessoas

Biscoito

Para testar o sistema de segurança de alguém; roubar dados

Representante de vendas

Afirmar que representa toda a Europa, não apenas Andorra

Corporação

Para descobrir o plano de marketing estratégico de um concorrente

Ex-funcionário

Para se vingar por ser despedido

Contador
Para desviar dinheiro de uma empresa
Corretor da bolsa
Para negar uma promessa feita a um cliente por e-mail
Ladrão de identidade
Para roubar números de cartão de crédito para venda
Governo
Para aprender os segredos militares ou industriais de um inimigo
Terrorista
Para roubar segredos da guerra biológica

Figura 8-1. Algumas pessoas que podem causar problemas de segurança e por quê.
Problemas de segurança de rede podem ser divididos aproximadamente em quatro áreas interligadas: sigilo, autenticação, não-repúdio e controle de integridade.
Sigilo, também chamado de confidencialidade, tem a ver com manter as informações fora do maozinhas sujas de usuários não autorizados. Isso é o que geralmente vem à mente quando as pessoas pensam sobre segurança de rede. A autenticação trata de determinar com quem você está falando antes de revelar informações confidenciais ou entrar em um negócio. O não-repúdio lida com assinaturas: como você prova que seu cliente realmente fez um pedido eletrônico de dez milhões de doohickeys canhotos a 89 centavos cada quando ele afirma que o preço era de 69 centavos? Ou talvez ele alega ele nunca fez nenhum pedido. Finalmente, o controle de integridade tem a ver com como você pode certifique-se de que a mensagem que você recebeu foi realmente enviada e não algo que um adversário malicioso modificou em trânsito ou inventou.
Todas essas questões (sigilo, autenticação, não-repúdio e integridade control) também ocorrem em sistemas tradicionais, mas com algumas diferenças significativas. No-tegridade e sigilo são alcançados usando correio registrado e documentos de bloqueio acima. Roubar o trem do correio é mais difícil agora do que nos dias de Jesse James. Além disso, as pessoas geralmente podem dizer a diferença entre um documento original em papel e uma fotocópia, e muitas vezes é importante para eles. Como teste, faça uma fotocópia de um cheque válido. Experimente descontar o cheque original no seu banco na segunda-feira. Agora tente descontar a fotocópia do cheque na terça-feira. Observe a diferença no comportamento do banco. Com cheques eletrônicos, o original e a cópia são indistintos decifrável. Pode demorar um pouco para os bancos aprenderem como lidar com isso. As pessoas autenticam outras pessoas por vários meios, incluindo o reconhecimento seus rostos, vozes e caligrafia. A prova de assinatura é tratada por assinaturas em papel timbrado, selos em relevo e assim por diante. A violação geralmente pode ser detectada por

Página 789

765

especialistas em escrita, tinta e papel. Nenhuma dessas opções estão disponíveis eletronicamente. Claramente, outras soluções são necessárias.
Antes de entrar nas próprias soluções, vale a pena gastar alguns momentos, considerando onde a segurança da rede da pilha do protocolo pertence. Lá provavelmente não é um único lugar. Cada camada tem algo a contribuir. Na camada física, escuta telefônica pode ser evitada fechando linhas de transmissão (ou melhor ainda, fibras ópticas) em tubos selados contendo um gás inerte em alta pressão. Qualquer tentativa de perfurar um tubo irá liberar algum gás, reduzindo a pressão e gerando um alarme. Alguns sistemas militares usam essa técnica.
Na camada de enlace de dados, os pacotes em uma linha ponto a ponto podem ser criptografados à medida que saem de uma máquina e descriptografados à medida que entram em outra. Todos os detalhes podem ser tratada na camada de enlace de dados, com as camadas superiores alheias ao que está acontecendo. Esta solução falha quando os pacotes têm que atravessar vários roteadores, entretanto-er, porque os pacotes precisam ser descriptografados em cada roteador, deixando-os vulneráveis a ataques de dentro do roteador. Além disso, não permite que algumas sessões sejam protegidas (por exemplo, aquelas que envolvem compras online com cartão de crédito) e outras não.

No entanto, a **criptografia de link**, como esse método é chamado, pode ser adicionada a qualquer rede

funcionam facilmente e são frequentemente úteis.

Na camada de rede, firewalls podem ser instalados para manter pacotes bons e ruins pacotes para fora. A segurança IP também funciona nesta camada.

Na camada de transporte, conexões inteiras podem ser criptografadas de ponta a ponta, ou seja, processo para processar. Para segurança máxima, é necessária segurança de ponta a ponta.

Finalmente, questões como autenticação de usuário e não-repúdio só podem ser tratada na camada de aplicação.

Visto que a segurança não se encaixa perfeitamente em nenhuma camada, ela não se encaixa em nenhum capítulo deste livro. Por isso, avalia seu próprio capítulo.

Embora este capítulo seja longo, técnico e essencial, também é quase irrelevante para o momento. É bem documentado que a maioria das falhas de segurança em bancos, por exemplo, são devido a procedimentos de segurança negligentes e funcionários incompetentes, numerosos

bugs de implementação que permitem invasões remotas por usuários não autorizados, e assim chamados de ataques de engenharia social, onde os clientes são levados a revelar seus Detalhes da conta. Todos esses problemas de segurança são mais prevalentes do que inteligentes criminosos grampeando linhas telefônicas e, em seguida, decodificando mensagens criptografadas. Se uma pessoa

pode entrar em uma agência aleatória de um banco com um talão de caixa eletrônico que encontrou na rua

alegando ter esquecido seu PIN e obter um novo no local (em nome de boas relações com o cliente), toda a criptografia do mundo não impedirá o abuso.

A este respeito, o livro de Ross Anderson (2008a) é uma verdadeira revelação, uma vez que documenta

mentos centenas de exemplos de falhas de segurança em vários setores, quase todos deles devido ao que pode ser educadamente chamado de práticas de negócios desleixadas ou inatten-a segurança. No entanto, a base técnica sobre a qual o e-commerce é construída quando todos esses outros fatores são bem executados é a criptografia.

Exceto para a segurança da camada física, quase toda a segurança da rede é baseada em princípios criptográficos. Por este motivo, começaremos nosso estudo de segurança por

Página 790

766

SEGURANÇA DE REDE INDIVÍDUO. 8

examinar a criptografia em alguns detalhes. Na seção 8.1, veremos alguns dos princípios básicos. Na seção 8-2 a Seç. 8-5, vamos examinar algumas das algoritmos damentais e estruturas de dados usados em criptografia. Então vamos examine em detalhes como esses conceitos podem ser usados para obter segurança em redes. Concluiremos com algumas reflexões sobre tecnologia e sociedade.

Antes de começar, cabe um último pensamento: o que não está coberto. Nós temos tentou se concentrar em problemas de rede, em vez de sistema operacional e aplicativo questões, embora a linha seja frequentemente difícil de traçar. Por exemplo, não há nada aqui sobre autenticação de usuário usando biometria, segurança de senha, estouro de buffer em-tacks, cavalos de Tróia, spoofing de login, injeção de código, como cross-site scripting, víardis, vermes e semelhantes. Todos esses tópicos são abordados detalhadamente no cap. 9 de *Sistemas operacionais modernos* (Tanenbaum, 2007). O leitor interessado é encaminhado a esse livro para os aspectos de sistemas de segurança. Agora vamos começar nossa jornada.

8.1 CRIPTOGRAFIA

A **criptografia** vem das palavras gregas para "escrita secreta". Tem uma longa e colorida história que remonta a milhares de anos. Nesta seção, iremos apenas esboçar alguns dos destaques, como informações básicas para o que se segue. Para uma história completa da criptografia, o livro de Kahn (1995) é recomendado lendo. Para um tratamento abrangente de segurança moderna e criptográfica al-

goritmos, protocolos e aplicativos e material relacionado, consulte Kaufman et al. (2002). Para uma abordagem mais matemática, consulte Stinson (2002). Por menos abordagem matemática, ver Burnett e Paine (2001).

Os profissionais fazem uma distinção entre cifras e códigos. Uma **cifra** é um transformação de caractere por caractere ou bit a bit, sem levar em conta a linguagem estrutura da mensagem. Em contraste, um **código** substitui uma palavra por outra palavra ou símbolo. Os códigos não são mais usados, embora tenham uma história gloriosa histórica. O código de maior sucesso já desenvolvido foi usado pelas forças armadas dos EUA durante a Segunda Guerra Mundial no Pacífico. Eles simplesmente tinham índios Navajo conversando com

uns aos outros usando palavras Navajo específicas para termos militares, por exemplo *chay-dagahi-nail-tsaidi* (literalmente: assassino de tartaruga) para arma antitanque. The Navajo lan-a linguagem é altamente tonal, extremamente complexa e não tem forma escrita. E não um pessoa solteira no Japão sabia nada sobre isso.

Em setembro de 1945, o *San Diego Union* descreveu o código dizendo " Para três anos, onde quer que os fuzileiros navais desembarcassem, os japoneses recebiam uma porção de estranhos

ruídos gorgolejantes intercalados com outros sons que lembram o chamado de um tibetano monge e o som de uma garrafa de água quente sendo esvaziada. " Os japoneses nunca quebrou o código e muitos codificadores Navajo receberam altas honras militares pelo serviço extraordinário e bravura. O fato de que os EUA quebraram os japoneses código, mas os japoneses nunca quebraram o código Navajo desempenhou um papel crucial no Vitórias americanas no Pacífico.

Página 791

SEC. 8,1
CRIPTOGRAFIA

767

8.1.1 Introdução à criptografia

Historicamente, quatro grupos de pessoas usaram e contribuíram para a arte de criptografia: os militares, o corpo diplomático, diaristas e amantes. Destes, o militar teve o papel mais importante e moldou o campo ao longo do séculos. Dentro das organizações militares, as mensagens a serem criptografadas têm tradição aliado a funcionários mal pagos e de baixo nível para criptografar e transmitir sion. O grande volume de mensagens impediu que este trabalho fosse feito por um poucos especialistas de elite.

Até o advento dos computadores, uma das principais restrições à criptografia tinha sido a capacidade do secretário de código de realizar as transformações necessárias, frequentemente em um campo de batalha com pouco equipamento. Uma restrição adicional foi a dificuldade em mudar rapidamente de um método criptográfico para outro um, pois isso envolve a reciclagem de um grande número de pessoas. No entanto, o perigo de um secretário de código sendo capturado pelo inimigo tornou essencial ser capaz de mude o método criptográfico instantaneamente se necessário. Estes conflitos conflitantes as peculiaridades deram origem ao modelo da Fig. 8-2.

Encriptação
método, E
Passiva
intruso
somente
escuta
Ativo
intruso
pode alterar
mensagens
Texto simples, P
Texto simples, P
Decifrar
método, D
Encriptação
chave, K
Decifrar
chave, K
Texto cifrado, C = E k (P)

Intruso

Figura 8-2. O modelo de criptografia (para uma cifra de chave simétrica).

As mensagens a serem criptografadas, conhecidas como **texto simples**, são transformadas por um função que é parametrizada por uma **chave**. A saída do processo de criptografia, conhecido como **texto cifrado**, é então transmitido, geralmente por mensageiro ou rádio. Nós as-suponha que o inimigo, ou **intruso**, ouça e copie com precisão o completo texto cifrado. No entanto, ao contrário do destinatário pretendido, ele não sabe o que A chave de descriptografia é e, portanto, não pode descriptografar o texto cifrado facilmente. Às vezes, o truder não pode apenas ouvir o canal de comunicação (intruso passivo), mas pode também gravar mensagens e reproduzi-las mais tarde, injetar suas próprias mensagens ou modificar fy mensagens legítimas antes de chegarem ao receptor (intruso ativo). A arte de

Página 792

768

SEGURANÇA DE REDE
INDIVÍDUO. 8

quebrar cifras, conhecido como **criptoanálise**, e a arte de concebê-los (cryptograph) são conhecidos coletivamente como **criptologia**.

Muitas vezes será útil ter uma notação para relacionar texto simples, texto cifrado e chaves. Usaremos $C = E_K(P)$ para significar que a criptografia do texto simples P usando chave K dá o texto cifrado C . Da mesma forma, $P = D_K(C)$ representa a descriptografia de C para obter o texto simples novamente. Segue-se então que

$$D_K(E_K(P)) = P$$

Esta notação sugere que E e D são apenas funções matemáticas, que eles estão. A única parte complicada é que ambos são funções de dois parâmetros, e nós escreveu um dos parâmetros (a chave) como um subscrito, ao invés de um argumento ment, para distingui-lo da mensagem.

Uma regra fundamental da criptografia é que se deve assumir que a criptografia analista conhece os métodos usados para criptografia e descriptografia. Em outras palavras, o criptanalista sabe como o método de criptografia, E , e descriptografia, D , de A Fig. 8-2 funciona em detalhes. A quantidade de esforço necessária para inventar, testar e instalar um novo algoritmo toda vez que o método antigo é comprometido (ou pensado para ser comprometido) sempre tornou impraticável manter o algoritmo de criptografia segredo. Pensar que é segredo quando não faz mais mal do que bem.

É aqui que entra a chave. A chave consiste em uma string (relativamente) curta que seleciona uma das muitas criptografias potenciais. Em contraste com o método geral, que só pode ser alterado a cada poucos anos, a chave pode ser alterada tão frequentemente quanto requeridos. Assim, nosso modelo básico é um método geral estável e conhecido publicamente parametrizado por uma chave secreta e facilmente alterada. A ideia de que o criptanalista

conhece os algoritmos e que o sigilo reside exclusivamente nas chaves é chamado **Princípio de Kerckhoff**, em homenagem ao criptógrafo militar flamengo Auguste Kerckhoff que o declarou pela primeira vez em 1883 (Kerckhoff, 1883). Assim, temos

Princípio de Kerckhoff: Todos os algoritmos devem ser públicos; apenas as chaves são secretas

O não sigilo do algoritmo não pode ser enfatizado o suficiente. Tentando manter o algoritmo em segredo, conhecido no mercado como **segurança por obscuridade**, nunca trabalho. Além disso, ao divulgar o algoritmo, o criptógrafo obtém consultoria gratuita de um grande número de criptologistas acadêmicos ansiosos para quebrar o sistema para que eles podem publicar artigos demonstrando como eles são inteligentes. Se muitos especialistas tentaram quebrar o algoritmo por um longo tempo após sua publicação e ninguém conseguiu necessário, provavelmente é bastante sólido.

Uma vez que o segredo real está na chave, seu comprimento é um grande problema de design. Considere

er uma fechadura de combinação simples. O princípio geral é que você insira dígitos em sequência. Todo mundo sabe disso, mas a chave é secreta. Um comprimento de chave de dois dígitos significa que existem 100 possibilidades. Um comprimento de chave de três dígitos significa 1000

possibilidades, e um comprimento de chave de seis dígitos significa um milhão. Quanto mais longa a chave, quanto maior for o **fator de trabalho com o qual** o criptanalista terá de lidar. O fator de trabalho para quebrar o sistema pela busca exaustiva do espaço-chave é exponencial no

Página 793

SEC. 8,1
CRIPTOGRAFIA

769

comprimento da chave. O sigilo vem de ter um algoritmo forte (mas público) e um longo chave. Para evitar que seu irmão mais novo leia seu e-mail, as chaves de 64 bits servem. Para uso comercial de rotina, pelo menos 128 bits devem ser usados. Para manter o grande governo Em caso de bloqueio, são necessárias chaves de pelo menos 256 bits, de preferência mais. Do ponto de vista do criptanalista, o problema da criptoanálise tem três principais variações. Quando ele tem uma quantidade de texto cifrado e nenhum texto simples, ele é confrontado com o problema **apenas do texto cifrado**. Os criptogramas que aparecem no A seção de quebra-cabeças dos jornais apresenta esse tipo de problema. Quando o criptanalista tem algum texto cifrado e texto simples, o problema é **conhecido como** problema de **texto simples**. Finalmente, quando o criptoanalista tem a capacidade de criptografar pedaços de texto simples de sua própria escolha, temos o problema do **texto simples escolhido**. Os criptogramas de jornais poderiam ser quebrados trivialmente se o criptanalista fosse permitido para fazer perguntas como " Qual é a criptografia de ABCDEFGHIJKL? " Os novatos no negócio de criptografia muitas vezes presumem que se uma cifra pode conter resistir a um ataque apenas de texto cifrado, é seguro. Essa suposição é muito ingênuca. No em muitos casos, o criptanalista pode fazer uma boa estimativa de partes do texto simples. Para Por exemplo, a primeira coisa que muitos computadores dizem quando você os chama é " login: ". Equipado com alguns pares de texto simples-texto cifrado, o trabalho do criptoanalista será vem muito mais fácil. Para obter segurança, o criptógrafo deve ser conservador e certifique-se de que o sistema é inquebrável, mesmo que seu oponente possa entrar criptografar quantidades arbitrárias de texto simples escolhido. Os métodos de criptografia foram historicamente divididos em duas categorias: substi-cifras de instrução e cifras de transposição. Agora vamos lidar com cada um desses resumidamente, como informações básicas para a criptografia moderna.

8.1.2 Cifras de Substituição

Em uma **cifra de substituição**, cada letra ou grupo de letras é substituído por outro carta ou grupo de letras para disfarçá-lo. Uma das mais antigas cifras conhecidas é a **Cifra de César**, atribuída a Júlio César. Com este método, *a* torna - se *D* , *b* seja-vem *E* , *c* torna-se *F* , ... , e *z* se torna *C* . Por exemplo, o **ataque** se torna *DWWDFN* . Em nossos exemplos, o texto simples será fornecido em letras minúsculas e texto cifrado em letras maiúsculas.

Uma ligeira generalização da cifra de César permite que o alfabeto do texto cifrado ser deslocado por *k* letras, em vez de sempre três. Neste caso, *k* torna-se uma chave para o método geral de alfabetos deslocados circularmente. A cifra de César pode ter enganou Pompeu, mas não enganou ninguém desde então.

A próxima melhoria é ter cada um dos símbolos no texto simples, digamos, o 26 letras para simplificar, mapear em alguma outra letra. Por exemplo, a B C D e F G H I J K L M N o p q R S T U V W x y Z

QWERTYUIOPASDFGHJKLZCVBNM

texto simples:

texto cifrado:

Página 794

770

SEGURANÇA DE REDE
INDIVÍDUO. 8

O sistema geral de substituição de símbolo por símbolo é chamado de **monoalfabético cifra de substituição**, com a chave sendo a string de 26 letras correspondente ao alfabeto completo. Para a chave fornecida, o *ataque de texto simples* seria transformado no texto cifrado *QZZQEA*.

À primeira vista, pode parecer um sistema seguro, porque embora o criptanalista conhece o sistema geral (substituição de letra por letra), ele não sabe qual dos $26! \sim 4 \times 10^{26}$ teclas possíveis em uso. Em contraste com o Cifra de César, tentar todos eles não é uma abordagem promissora. Mesmo a 1 nseg por solução, um milhão de chips de computador trabalhando em paralelo levaria 10.000 anos para tentar todas as chaves.

No entanto, dada uma quantidade surpreendentemente pequena de texto cifrado, a cifra pode ser quebrado facilmente. O ataque básico tira proveito das propriedades estatísticas de línguas naturais. Em inglês, por exemplo, *e* é a letra mais comum, seguida por *t, o, a, n, i*, etc. As combinações de duas letras mais comuns, ou **digramas**, são *th, in, er, re e an*. As combinações de três letras mais comuns, ou **trigramas**, são *o, ing, e, e ion*.

Um criptanalista tentando quebrar uma cifra monoalfabética começaria por contando as frequências relativas de todas as letras no texto cifrado. Então ele pode tentativamente, atribua o mais comum a *e* e o próximo mais comum a *t*.

Ele, então, olha os trigramas para encontrar um comum na forma *tXe*, que sugere fortemente que *X* é *h*. Da mesma forma, se o padrão *thYt* ocorre com frequência, o *Y* provavelmente significa *a*. Com essas informações, ele pode procurar um trígrama curricular da forma *aZW*, que é mais provável *e*. Fazendo suposições em letras, digramas e trigramas comuns e saber sobre os prováveis padrões de vogais e consoantes, o criptanalista constrói um texto simples provisório, letra por carta.

Outra abordagem é adivinhar uma palavra ou frase provável. Por exemplo, considere o seguinte texto cifrado de uma empresa de contabilidade (bloqueado em grupos de cinco caracteres):

CTBMN BYCTC BTJDS QXBNS GSTJC BTSWX CTQTZ CQVUJ
QJSGS TJQZZ MNQJS VLNSX VSZJU JDSTS JQUUS JUBXJ
DSKSU JSNTK BGAQJ ZBGYQ TLCTZ BNYBN QJSW

Uma palavra provável em uma mensagem de uma empresa de contabilidade é *financeira*. Usando nosso

conhecimento de que *financeiro* tem uma letra repetida (*i*), com outras quatro letras entre suas ocorrências, procuramos letras repetidas no texto cifrado neste espaçamento.

Encontramos 12 ocorrências nas posições 6, 15, 27, 31, 42, 48, 56, 66, 70, 71, 76 e 82.

No entanto, apenas dois deles, 31 e 42, têm a próxima letra (correspondendo a *n* em texto simples) repetido no local apropriado. Destes dois, apenas 31 tem também a *um* posicionado corretamente, então sabemos que o *financeiro* começa na posição 30. A partir disso ponto em diante, deduzir a chave é fácil usando as estatísticas de frequência para o inglês texto e procurando palavras quase completas para terminar.

Página 795

SEC. 8,1
CRIPTOGRAFIA

771

8.1.3 Cifras de Transposição

As cifras de substituição preservam a ordem dos símbolos do texto simples, mas disfarçam eles. As **cifras de transposição**, em contraste, reordenam as letras, mas não disfarçam eles. A Figura 8-3 descreve uma cifra de transposição comum, a transposição colunar.

Neste exemplo, MEGABUCK é a chave. O objetivo da chave é ordenar as colunas, com a coluna 1 sob a letra-chave mais próxima do início do alfabeto e assim por diante. O texto simples é escrito horizontalmente, em linhas, preenchido para preencher a matriz, se necessário. O texto cifrado é lido por colunas, começando com a coluna

cuja letra chave é a mais baixa.

MEGABUCK
7 4 5 1 2 8 3 6
por favor
Texto simples
Por favor, transfira um milhão de dólares para
myswissbankaccountsixtwotwo
Texto cifrado
AFLLSKSOSELAWAIATOSSCTCLNMOMANT
ESILYNTWRNNTSOWDPAEDOBUERIRICXB
anferon
emilhão
dólar
omyswiss
Bankacco
untsixtw
ottoabcd

Figura 8-3. Uma cifra de transposição.

Para quebrar uma cifra de transposição, o criptanalista deve primeiro estar ciente de que ele está lidar com uma cifra de transposição. Olhando para a frequência de E , T , A , O , I , N , etc., é fácil ver se eles se encaixam no padrão normal para texto simples. Se sim, a cifra é claramente uma cifra de transposição, porque em tal cifra cada letra a representa-self, mantendo a distribuição de frequência intacta.

A próxima etapa é adivinhar o número de colunas. Em muitos casos, um palavra ou frase provável pode ser adivinhada a partir do contexto. Por exemplo, supõe que nosso criptanalista suspeita que a frase de texto simples *milhões de dólares* ocorre em algum lugar da mensagem. Observe que os digramas *MO*, *IL*, *LL*, *LA*, *IR* e *OS* ocorrem cur no texto cifrado como resultado desta frase envolvendo. O texto cifrado a letra *O* segue a letra *M* do texto cifrado (ou seja, eles são verticalmente adjacentes na coluna 4) porque estão separados na frase provável por uma distância igual à chave comprimento. Se uma chave de comprimento sete foi usada, os dígitos *MD*, *IO*, *LL*, *LL*, *IA*, *OR* e *NS* teriam ocorrido em seu lugar. Na verdade, para cada comprimento de chave, um diferente conjunto de digramas é produzido no texto cifrado. Ao caçar as várias possibilidades laços, o criptanalista pode frequentemente determinar facilmente o comprimento da chave.

Página 796

772

SEGURANÇA DE REDE
INDIVÍDUO. 8

A etapa restante é ordenar as colunas. Quando o número de colunas, k , é pequeno, cada um dos k ($k - 1$) pares de colunas pode ser examinado para ver se suas frequências do digrama correspondem às do texto simples em inglês. O par com o melhor a correspondência é considerada como corretamente posicionada. Agora, cada uma das colunas restantes

é tentado como o sucessor deste par. A coluna cujo digrama e suas frequências de grama fornecem a melhor correspondência é provisoriamente assumido como correto. o

a próxima coluna é encontrada da mesma maneira. Todo o processo é continuado até um ponto pedido potencial for encontrado. As chances são de que o texto simples seja reconhecível em neste ponto (por exemplo, se ocorrer *milloin*, é claro qual é o erro).

Algumas cifras de transposição aceitam um bloco de entrada de comprimento fixo e produzem um bloco de saída de comprimento fixo. Essas cifras podem ser completamente descritas por uma lista informando a ordem em que os caracteres devem ser produzidos. Por exemplo, a cifra da Figura 8-3 pode ser vista como uma cifra de bloco de 64 caracteres. Sua saída é 4, 12, 20, 28, 36, 44, 52, 60, 5, 13, . . . , 62. Em outras palavras, o quarto caractere de entrada é o primeiro a ser produzido, seguido pelo décimo segundo, *f* e assim por diante.

8.1.4 Pads de uso único

Construir uma cifra inquebrável é realmente muito fácil; a técnica tem conhecido há décadas. Primeiro, escolha uma string de bits aleatória como a chave. Então converta o texto simples em uma string de bits, por exemplo, usando sua representação ASCII. Por fim, calcule o XOR (OU exclusivo) dessas duas strings, bit a bit. Lá, o texto cifrado de pesquisa não pode ser quebrado porque em uma amostra suficientemente grande de

texto cifrado, cada letra ocorrerá com a mesma frequência, assim como cada digrama, cada trigrama e assim por diante. Este método, conhecido como **pad de uso único**, é imune a todas as pressões

ataques atuais e futuros, não importa quanto poder computacional o intruso tenha.

A razão deriva da teoria da informação: simplesmente não há informações no mensagem porque todos os textos simples possíveis do comprimento fornecido são igualmente prováveis.

Um exemplo de como os pads descartáveis são usados é dado na Fig. 8-4. Primeiro, mes-sábio 1, " Eu te amo " é convertido para ASCII de 7 bits. Então, um pad de uso único, pad 1, é escolhido e XORed com a mensagem para obter o texto cifrado. Um criptanalista poderia experimente todos os pads de uso único possíveis para ver qual texto simples saiu para cada um. Para

exemplo, o pad descartável listado como pad 2 na figura pode ser tentado, resultando em texto simples 2, " Elvis vive ", que pode ou não ser plausível (um assunto além âmbito deste livro). Na verdade, para cada texto simples ASCII de 11 caracteres, há um pad único que o gera. Isso é o que queremos dizer ao dizer que não há informações mação no texto cifrado: você pode obter qualquer mensagem com o comprimento correto a partir dele.

Os pads descartáveis são ótimos na teoria, mas têm uma série de desvantagens na prática tice. Para começar, a chave não pode ser memorizada, então tanto o remetente quanto o receptor deve levar consigo uma cópia escrita. Se qualquer um estiver sujeito a captura, por escrito chaves são claramente indesejáveis. Além disso, a quantidade total de dados que podem ser transmitido é limitado pela quantidade de chave disponível. Se o espião ficar rico e descobre uma grande quantidade de dados, ele pode se ver incapaz de transmiti-los de volta para

Página 797

SEC. 8,1
CRIPTOGRAFIA

773

Mensagem 1:
1001001 0100000 1101100 1101111 1110110 1100101 0100000 1111001 1101111 1110101 0101110
Bloco 1:
1010010 1001011 1110010 1010101 1010010 1100011 0001011 0101010 1010111 1100110 0101011
Texto cifrado:
0011011 1101011 0011110 0111010 0100100 0000110 0101011 1010011 0111000 0010011 0000101
Pad 2:
1011110 0000111 1101000 1010011 1010111 0100110 1000111 0111010 1001110 1110110 1110111
Texto simples 2:
1000101 1101100 1110110 1101001 1110011 0100000 1101100 1101001 1110110 1100101 1110011

Figura 8-4. O uso de um teclado único para criptografia e a possibilidade de obter ting qualquer texto simples possível do texto cifrado pelo uso de algum outro bloco. sede porque a chave foi usada. Outro problema é a sensibilidade tipo do método para caracteres perdidos ou inseridos. Se o remetente e o destinatário saírem de sincronização, todos os dados a partir de então aparecerão truncados. Com o advento dos computadores, o teclado único pode potencialmente se tornar prático para algumas aplicações. A fonte da chave pode ser um DVD especial que contém vários gigabytes de informação e, se transportado em um filme de DVD e prefixado por alguns minutos de vídeo, nem seria suspeito. Do claro, em velocidades de rede gigabit, ter que inserir um novo DVD a cada 30 segundos poderia torna-se tedioso. E os DVDs devem ser transportados pessoalmente do remetente para o receptor antes que qualquer mensagem possa ser enviada, o que reduz muito sua prática Utilitário.

Criptografia quântica

Curiosamente, pode haver uma solução para o problema de como transmitir o one-time pad pela rede, e vem de uma fonte muito improvável: quant-uma mecânica. Esta área ainda é experimental, mas os testes iniciais são promissores. Se isso pode ser aperfeiçoado e tornar-se eficiente, virtualmente toda criptografia acabará ser feito usando pads de uso único, uma vez que são comprovadamente seguros. Abaixo, iremos resumir

Explique como esse método, a **criptografia quântica**, funciona. Em particular, nós irá descrever um protocolo chamado **BB84** após seus autores e ano de publicação (Bennet e Brassard, 1984).

Suponha que um usuário, Alice, deseja estabelecer um pad de uso único com um segundo

usuário, Bob. Alice e Bob são chamados de **diretores**, os personagens principais de nossa história. Por exemplo, Bob é um banqueiro com quem Alice gostaria de fazer negócios. Os nomes "Alice" e "Bob" têm sido usados para os diretores em praticamente todos os papéis e livros sobre criptografia desde que Ron Rivest os apresentou, muitos anos atrás (Rivest et al., 1978). Os criptógrafos amam a tradição. Se fossemos usar "Andy" e "Barbara" como os diretores, ninguém acreditaria em nada nisso no capítulo. Que assim seja.

Se Alice e Bob pudessem estabelecer um bloco único, eles poderiam usá-lo para se comunicar late com segurança. A questão é: como eles podem estabelecer isso sem previamente trocando DVDs? Podemos supor que Alice e Bob estão nas extremidades opostas

Página 798

774

SEGURANÇA DE REDE

INDIVÍDUO. 8

de uma fibra óptica pela qual podem enviar e receber pulsos de luz. Contudo, um intrépido intruso, Trudy, pode cortar a fibra para emendar em uma torneira ativa. Trudy sabe ler todos os bits enviados em ambas as direções. Ela também pode enviar mensagens falsas em ambos

instruções. A situação pode parecer desesperadora para Alice e Bob, mas quântica a criptografia pode lançar uma nova luz sobre o assunto.

A criptografia quântica é baseada no fato de que a luz vem em pequenos pacotes chamados **fôtons**, que têm algumas propriedades peculiares. Além disso, a luz pode ser polarizada por passar por um filtro polarizador, fato bem conhecido por ambos os usuários de óculos de sol e fotógrafos. Se um feixe de luz (ou seja, um fluxo de fotônulos) é passado por um filtro polarizador, todos os fôtons que emergem dele serão polarizados na direção do eixo do filtro (por exemplo, verticalmente). Se o feixe é agora passado por um segundo filtro polarizador, a intensidade da luz emergindo de

o segundo filtro é proporcional ao quadrado do cosseno do ângulo entre os machados. Se os dois eixos são perpendiculares, nenhum fôton passa. O absoluto a orientação dos dois filtros não importa; apenas o ângulo entre seus eixos contagens.

Para gerar um bloco de uso único, Alice precisa de dois conjuntos de filtros polarizadores. Defina um

consiste em um filtro vertical e um filtro horizontal. Esta escolha é chamada de **retilínea**. Uma base (plural: bases) é apenas um sistema de coordenadas. O segundo conjunto de filtros é o mesmo, exceto girado 45 graus, então um filtro é executado a partir do canto inferior esquerdo

para a direita superior e o outro filtro vai da esquerda superior para a direita inferior.

Essa escolha é chamada de **base diagonal**. Assim, Alice tem duas bases, que ela pode inserir rapidamente em sua viga à vontade. Na realidade, Alice não tem quatro

filtros, mas um cristal cuja polarização pode ser comutada eletricamente para qualquer um dos

quatro direções permitidas em grande velocidade. Bob tem o mesmo equipamento que Alice.

O fato de Alice e Bob terem cada um duas bases disponíveis é essencial para a quantum criptografia.

Para cada base, Alice agora atribui uma direção como 0 e a outra como 1. No exemplo apresentado abaixo, presumimos que ela escolhe vertical como 0 e horizontal para ser 1. Independentemente, ela também escolhe inferior esquerdo para superior direito como 0 e superior

da esquerda para a direita inferior como 1. Ela envia essas opções para Bob como texto simples.

Agora Alice escolhe um teclado de uso único, por exemplo, baseado em um número aleatório gerador (um assunto complexo por si só). Ela transfere aos poucos para Bob, escolhendo uma de suas duas bases aleatoriamente para cada bit. Para enviar um pouco, sua arma de fôtons emite

um fôton polarizado adequadamente para a base que ela está usando para aquele bit. Para exemplificar, ela pode escolher bases de diagonal, retilínea, retilínea, diagonal, retilínea, etc. Para enviar seu bloco único de 1001110010100110 com essas bases,

SEC. 8,1
CRIPTOGRAFIA

775

Trudy
almofada

(g)

x

0

x

1

x

x

x

?

1

x

?

?

0

x

?

0

1

0

1

1

0

0

1

x

Não Sim Não Sim Não Não Sim Sim Não Sim Sim Sim Não Sim Não

Mordeu

número

Dados

Trudy

bases

(f)

1-

Tempo

almofada

(e)

Corrigir

base?

(d)

o que

Prumo

pega

(c)

Bob's

bases

(b)

o que

Alice

envia

(uma)

1

0

0

1

1

0

0

1

1

0

0

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

Figura 8-5. Um exemplo de criptografia quântica.

porque se um fóton atinge um filtro polarizado em 45 graus para sua própria polarização, ele salta aleatoriamente para a polarização do filtro ou para uma polarização perpendicular ao filtro, com igual probabilidade. Esta propriedade dos fótons é fundamental para mecânica quântica. Assim, alguns dos bits estão corretos e alguns são aleatórios, mas Bob não sabe quais são quais. Os resultados de Bob estão representados na Fig. 8-5 (c). Como Bob descobre quais bases ele acertou e quais errou? Ele simplesmente diz a Alice qual base ele usou para cada bit em texto simples e ela diz a ele quais estão certas e quais estão erradas no texto simples, conforme mostrado na Figura 8-5 (d). De esta informação, ambos podem construir uma string de bits a partir das suposições corretas, como mostrado na Fig. 8-5 (e). Em média, esta string de bits terá metade do comprimento do string de bits original, mas como ambas as partes sabem disso, podem usá-la como um bloco de uso único.

Tudo que Alice precisa fazer é transmitir uma string de bits um pouco mais do que o dobro do desejado

comprimento, e ela e Bob terão um bloco único do comprimento desejado. Feito.

Mas espere um minuto. Esquecemos Trudy. Suponha que ela esteja curiosa sobre o que Alice tem a dizer e corta a fibra, inserindo seu próprio detector e transmissor.

Infelizmente para ela, ela não sabe qual base usar para cada fóton eitther. O melhor que ela pode fazer é escolher um aleatoriamente para cada fóton, assim como Bob faz.

Um exemplo de suas escolhas é mostrado na Figura 8-5 (f). Quando Bob relatar mais tarde (em texto simples) quais bases ele usou e Alice diz a ele (em texto simples) quais são

Página 800

776

SEGURANÇA DE REDE
INDIVÍDUO. 8

correto, Trudy agora sabe quando acertou e quando acertou. No Fig. 8-5, ela acertou para os bits 0, 1, 2, 3, 4, 6, 8, 12 e 13. Mas ela sabe de A resposta de Alice na Fig. 8-5 (d) que apenas os bits 1, 3, 7, 8, 10, 11, 12 e 14 fazem parte do o bloco de uso único. Para quatro desses bits (1, 3, 8 e 12), ela acertou e capturou o bit correto. Para os outros quatro (7, 10, 11 e 14), ela adivinhou errado e não sabe o bit transmitido. Assim, Bob sabe que o teclado único começa com 01011001, da Fig. 8-5 (e), mas tudo o que Trudy tem é 01? 1 ?? 0 ?, da Fig. 8-5 (g).

Claro, Alice e Bob estão cientes de que Trudy pode ter capturado parte de seu bloco de uso único, então eles gostariam de reduzir as informações que Trudy tem. Eles pode fazer isso realizando uma transformação nele. Por exemplo, eles podem dividir o bloco único em blocos de 1024 bits, quadrando cada um para formar um bloco de 2048 bits número e use a concatenação desses números de 2048 bits como o teclado único.

Com seu conhecimento parcial da string de bits transmitida, Trudy não tem como gerar erate seu quadrado e então não tem nada. A transformação da única vez original para um diferente que reduz o conhecimento de Trudy é chamado de **amplificação de privacidade**. Na prática, transformações complexas das quais cada bit de saída depende em cada bit de entrada são usados em vez de quadratura.

Pobre Trudy. Ela não só não tem ideia do que é o bloco de uso único, mas também presença também não é segredo. Afinal, ela deve retransmitir cada bit recebido para Bob

para fazê-lo pensar que está falando com Alice. O problema é que o melhor que ela pode fazer é transmitir o qubit que ela recebeu, usando a polarização que ela usou para recebê-lo, e cerca da metade das vezes ela estará errada, causando muitos erros no bloco único de Bob. Quando Alice finalmente começa a enviar dados, ela os codifica usando um forward pesado código de correção de erros. Do ponto de vista de Bob, um erro de 1 bit no teclado único é o mesmo que um erro de transmissão de 1 bit. De qualquer forma, ele entendeu a parte errada. E se há correção de erro direta suficiente, ele pode recuperar a mensagem original apesar de todos os erros, mas ele pode facilmente contar quantos erros foram corrigidos. E se este número é muito mais do que a taxa de erro esperada do equipamento, ele sabe que Trudy bateu na linha e pode agir de acordo (por exemplo, diga a Alice para mudar para uma estação de rádio, chame a polícia, etc.). Se Trudy tivesse uma maneira de clonar um fóton, então ela tinha um fóton para inspecionar e um fóton idêntico para enviar a Bob, ela poderia evitar detecção, mas no momento nenhuma maneira de clonar um fóton perfeitamente é conhecido. E mesmo se

Trudy poderia clonar fótons, o valor da criptografia quântica para estabelecer um os blocos de tempo não seriam reduzidos.

Embora a criptografia quântica tenha demonstrado operar em distâncias de Com 60 km de fibra, o equipamento é complexo e caro. Ainda assim, a ideia prometeu ise. Para obter mais informações sobre criptografia quântica, consulte Mullins (2002).

8.1.5 Dois princípios criptográficos fundamentais

Embora iremos estudar muitos sistemas criptográficos diferentes nas páginas adiante, dois princípios subjacentes a todos eles são importantes para entender. Pagamento atenção. Você os violar por sua conta e risco.

Página 801

SEC. 8,1
CRIPTOGRAFIA
777

Redundância

O primeiro princípio é que todas as mensagens criptografadas devem conter algum redundancy, isto é, informações não necessárias para entender a mensagem. Um exemplo pode deixar claro por que isso é necessário. Considere uma empresa de mala direta, a Couch Potato (TCP), com 60.000 produtos. Pensando que eles estão sendo muito eficientes suficiente, os programadores do TCP decidem que as mensagens de ordenação devem consistir em um 16-

byte nome do cliente seguido por um campo de dados de 3 bytes (1 byte para a quantidade e 2 bytes para o número do produto). Os últimos 3 bytes devem ser criptografados usando um chave longa conhecida apenas pelo cliente e pelo TCP.

A primeira vista, isso pode parecer seguro e, de certa forma, é porque intrusos passivos não pode descriptografar as mensagens. Infelizmente, ele também tem uma falha fatal que o torna sem utilidade. Suponha que um funcionário demitido recentemente queira punir a TCP por demissão dela. Antes de sair, ela leva a lista de clientes com ela. Ela trabalha através a noite escrevendo um programa para gerar pedidos fictícios usando clientes reais nomes. Como ela não tem a lista de chaves, ela apenas coloca números aleatórios em os últimos 3 bytes e envia centenas de pedidos para o TCP.

Quando essas mensagens chegam, o computador do TCP usa o nome do cliente para localize a chave e descriptografe a mensagem. Infelizmente para o TCP, quase todos os 3- a mensagem de byte é válida, então o computador começa a imprimir as instruções de envio. Embora possa parecer estranho para um cliente pedir 837 conjuntos de balanços infantis ou 540 sandboxes, pelo que o computador sabe, o cliente pode estar planejando abrir uma rede de parques infantis franqueados. Desta forma, um intruso ativo (o ex-funcionário) pode causar uma quantidade enorme de problemas, mesmo que ela não possa aguentar as mensagens que seu computador está gerando.

Este problema pode ser resolvido adicionando redundância a todas as mensagens. Por exemplo, se as mensagens de pedido forem estendidas para 12 bytes, os primeiros 9 dos quais devem

ser zeros, este ataque não funciona mais porque o ex-funcionário não pode mais gerar erar um grande fluxo de mensagens válidas. A moral da história é que todas as mensagens deve conter redundância considerável para que os intrusos ativos não possam enviar aleatoriamente lixo e fazer com que seja interpretado como uma mensagem válida.

No entanto, adicionar redundância torna mais fácil para os criptanalistas quebrar mensagens sábios. Suponha que o negócio de mala direta seja altamente competitivo e o The Couch O principal concorrente do Potato, The Sofa Tuber, adoraria saber quantos sandboxes que o TCP está vendendo, conecta-se à linha telefônica do TCP. No esquema original com

Mensagens de 3 bytes, a criptoanálise era quase impossível porque depois de adivinhar um chave, o criptanalista não tinha como saber se estava certo porque quase cada mensagem era tecnicamente legal. Com o novo esquema de 12 bytes, é fácil para

o criptanalista para distinguir uma mensagem válida de uma inválida. Assim, temos

Princípio criptográfico 1: as mensagens devem conter alguma redundância

Em outras palavras, ao descriptografar uma mensagem, o destinatário deve ser capaz de dizer se é válido simplesmente inspecionando a mensagem e talvez realizando um

Página 802

778

SEGURANÇA DE REDE

INDIVÍDUO. 8

computação simples. Esta redundância é necessária para evitar que intrusos ativos de enviar lixo e enganar o receptor para que ele descriptografe o lixo e atue no " texto simples ". No entanto, esta mesma redundância torna muito mais fácil para sive intrusos para quebrar o sistema, então há alguma tensão aqui. Além disso, o redundância nunca deve ser na forma de n 0s no início ou no final de uma mensagem, uma vez que executar essas mensagens por meio de alguns algoritmos criptográficos dá mais resultados previsíveis, facilitando o trabalho dos criptanalistas. Um polinômio CRC é muito melhor do que uma sequência de 0s, pois o receptor pode facilmente verificá-lo, mas gera mais trabalho para o criptanalista. Melhor ainda é usar um hash criptográfico, um con. Exceto que exploraremos mais tarde. Por enquanto, pense nele como um CRC melhor. Voltando à criptografia quântica por um momento, também podemos ver como a dundância desempenha um papel aí. Devido à interceptação dos fôtons por Trudy, alguns bits no teclado único de Bob estarão errados. Bob precisa de alguma redundância na próximas mensagens para determinar se os erros estão presentes. Uma forma muito grosseira de re- dundância está repetindo a mensagem duas vezes. Se as duas cópias não forem idênticas, Bob sabe que a fibra é muito barulhenta ou alguém está adulterando o transmissão. Claro, enviar tudo duas vezes é um exagero; um Hamming ou O código Reed-Solomon é uma forma mais eficiente de detectar e corrigir erros. Mas deve ficar claro que alguma redundância é necessária para distinguir uma mensagem válida sage de uma mensagem inválida, especialmente diante de um intruso ativo.

Frescor

O segundo princípio criptográfico é que devem ser tomadas medidas para garantir que cada mensagem recebida pode ser verificada como recente, ou seja, enviada muito recentemente. Esta medida é necessária para evitar que intrusos ativos reproduzam mensagens antigas. Se nenhuma dessas medidas fosse tomada, nosso ex-funcionário poderia explorar o TCP's linha telefônica e apenas continue repetindo mensagens válidas enviadas anteriormente. Portanto, *Princípio criptográfico 2: algum método é necessário para impedir ataques de repetição* Uma dessas medidas é incluir em cada mensagem um carimbo de data / hora válido apenas para, digamos, 10 segundos. O receptor pode então manter as mensagens por cerca de 10 segundos e compare as mensagens recém-chegadas às anteriores para filtrar as duplicatas. Mes- sábios com mais de 10 segundos podem ser descartados, uma vez que quaisquer replays enviados por mais de 10 segundos depois, será rejeitado como muito antigo. Medidas diferentes de timestamps irão ser discutido mais tarde.

8.2 ALGORITMOS DE CHAVE SIMÉTRICA

A criptografia moderna usa as mesmas idéias básicas da criptografia tradicional (transposição e substituição), mas sua ênfase é diferente. Tradicionalmente, cryptographers usaram algoritmos simples. Hoje em dia, o inverso é verdadeiro: o objeto

Página 803

SEC. 8,2

ALGORITMOS DE CHAVE SIMÉTRICA

779

é tornar o algoritmo de criptografia tão complexo e involuto que mesmo se o criptanalista adquire vastos montes de texto cifrado de sua própria escolha, ele irá não ser capaz de fazer nenhum sentido sem a chave.

A primeira classe de algoritmos de criptografia que estudaremos neste capítulo são chamadas ed **algoritmos de chave simétrica** porque eles usam a mesma chave para criptografia e descriptografia. A Fig. 8-2 ilustra o uso de um algoritmo de chave simétrica. Em particular lar, vamos nos concentrar em **cifras de bloco**, que usam um bloco de n bits de texto simples como insira e transforme-o usando a chave em um bloco de texto cifrado de n bits.

Algoritmos criptográficos podem ser implementados em qualquer hardware (para velocidade) ou software (para flexibilidade). Embora a maior parte do nosso tratamento diga respeito ao algoritmo

ritmos e protocolos, que são independentes da implementação real, alguns palavras sobre a construção de hardware criptográfico podem ser de interesse. Transposições e as substituições podem ser implementadas com circuitos elétricos simples. Figura 8-6 (a) mostra um dispositivo, conhecido como **P-box** (P significa permutação), usado para efetuar um transposição em uma entrada de 8 bits. Se os 8 bits forem designados de cima para baixo como 01234567, a saída desta caixa P particular é 36071245. Por meio de fiação final, uma P-box pode ser feita para realizar qualquer transposição e fazê-lo na prática exatamente a velocidade da luz, uma vez que nenhum cálculo está envolvido, apenas a propagação do sinal.

Este projeto segue o princípio de Kerckhoff: o atacante sabe que o general método está permutando os bits. O que ele não sabe é qual parte vai para onde.

S₁
S₂
P₁
P₄
P₃
P₂
S₃
S₄
S₅
S₆
S₇
S₈
Cifra de produto
(c)
S-box
Decodificador:
3
para
8
Codificador:
8
para
3
(b)
P-box
(uma)
S₉
S₁₀
S₁₁
S₁₂

Figura 8-6. Elementos básicos das cifras de produto. (a) P-box. (b) S-box. (c) Produto.

As substituições são realizadas por **S-boxes**, conforme mostrado na Fig. 8-6 (b). Neste exemplo, um texto simples de 3 bits é inserido e um texto cifrado de 3 bits é gerado. A entrada de 3 bits

seleciona uma das oito linhas que saem do primeiro estágio e a define como 1; todos outras linhas são 0. O segundo estágio é uma P-box. O terceiro estágio codifica a seleção

linha de entrada tem em binário novamente. Com a fiação mostrada, se os oito números octais 01234567 foram inseridos um após o outro, a sequência de saída seria 24506713. Em outras palavras, 0 foi substituído por 2, 1 foi substituído por 4, etc. Novamente, por fiação apropriada da caixa P dentro da caixa S, qualquer substituição pode ser acompanhada plished. Além disso, esse dispositivo pode ser construído em hardware para alcançar grandes velocidade, uma vez que codificadores e decodificadores têm apenas um ou dois (subnanosegundos) gate os atrasos e o tempo de propagação através da caixa P podem muito bem ser menores do que 1 picoseg.

Página 804

780

SEGURANÇA DE REDE INDIVÍDUO. 8

O verdadeiro poder desses elementos básicos só se torna aparente quando cas- cade uma série inteira de caixas para formar uma **cifra de produto**, como mostrado na Fig. 8-6 (c). Neste exemplo, 12 linhas de entrada são transpostas (ou seja, permutadas) pelo primeiro estágio (P_1). No segundo estágio, a entrada é dividida em quatro grupos de 3 bits, cada um dos que é substituído independentemente dos outros (S_1 a S_4). Este arranjo mostra um método de aproximação de uma S-box maior a partir de várias S-box menores. É útil porque pequenas S-boxes são práticas para uma implementação de hardware (por exemplo, uma S-box de 8 bits pode ser realizada como uma tabela de consulta de 256 entradas), mas grande S- caixas tornam-se difíceis de construir (por exemplo, uma caixa S de 12 bits teria uma necessidade mínima $2^{12} = 4096$ fios cruzados em seu estágio intermediário). Embora este método seja menos gen- eral, ainda é poderoso. Pela inclusão de um número suficientemente grande de estágios no cifra do produto, a saída pode ser uma função extremamente complicada da entrada.

Cifras de produto que operam em entradas de k - bits para produzir saídas de k - bits são muito comum. Normalmente, k é de 64 a 256. Uma implementação de hardware geralmente tem pelo menos 10 estágios físicos, em vez de apenas 7, como na Fig. 8-6 (c). Um implemento de software

a mentação é programada como um loop com pelo menos oito iterações, cada uma por formar substituições do tipo S-box em sub-blocos do bloco de dados de 64 a 256 bits, seguido por uma permutação que mistura as saídas das S-box. Freqüentemente há um permutação inicial especial e uma no final também. Na literatura, as itera- ções são chamadas de **rodadas**.

8.2.1 DES - O padrão de criptografia de dados

Em janeiro de 1977, o governo dos EUA adotou uma cifra de produto desenvolvida por IBM como seu padrão oficial para informações não classificadas. Esta cifra, **DES (dados Padrão de criptografia)**, foi amplamente adotado pela indústria para uso em segurança produtos. Não é mais seguro em sua forma original, mas em uma forma modificada é ainda útil. Vamos agora explicar como o DES funciona.

Um esboço do DES é mostrado na Fig. 8-7 (a). O texto simples é criptografado em blocos de 64 bits, resultando em 64 bits de texto cifrado. O algoritmo, que é parametrizado por um Chave de 56 bits, tem 19 estágios distintos. O primeiro estágio é uma transposição independente de chave

no texto simples de 64 bits. A última etapa é o inverso exato desta transposição ção. O estágio anterior ao último troca os 32 bits mais à esquerda com os da direita a maioria de 32 bits. Os 16 estágios restantes são funcionalmente idênticos, mas são parâme- tado por diferentes funções da chave. O algoritmo foi projetado para permitir que a criptografia seja feita com a mesma chave da criptografia, uma propriedade necessária em

qualquer algoritmo de chave simétrica. As etapas são executadas na ordem inversa.

A operação de um desses estágios intermediários é ilustrada na Fig. 8-7 (b).

Cada estágio leva duas entradas de 32 bits e produz duas saídas de 32 bits. A esquerda de fora

put é simplesmente uma cópia da entrada correta. A saída certa é o XOR bit a bit do entrada esquerda e uma função da entrada direita e a chave para este estágio, K_i . Bonita muito toda a complexidade do algoritmo reside nesta função.

Página 805

SEC. 8,2
ALGORITMOS DE CHAVE SIMÉTRICA

781

(b)
(uma)
Transposição inicial
Iteração 16
 $L_{i-1} \oplus f(R_{i-1}, K_i)$
 R_{i-1}
 L_{i-1}
Texto simples de 64 bits
Texto cifrado de 64 bits
32 bits
 L_i
32 bits
 R_i
Iteração 2
Iteração 1
56 bits
chave
Troca de 32 bits
Transposição inversa

Figura 8-7. O padrão de criptografia de dados. (a) Esboço geral. (b) Detalhe de uma iteração. O + circulado significa OU exclusivo.

A função consiste em quatro etapas, realizadas em sequência. Primeiro, um 48 bits número, E , é construído expandindo o 32-bit R_{i-1} de acordo com um fixo regra de transposição e duplicação. Em segundo lugar, E e K_i são submetidos a XOR juntos. Isto saída é então particionada em oito grupos de 6 bits cada, cada um dos quais é alimentado em um S-box diferente. Cada uma das 64 entradas possíveis para uma S-box é mapeada em um 4-saída de bits. Finalmente, esses bits 8×4 são passados por uma caixa P.

Em cada uma das 16 iterações, uma chave diferente é usada. Antes do algoritmo começa, uma transposição de 56 bits é aplicada à chave. Antes de cada iteração, o a chave é particionada em duas unidades de 28 bits, cada uma delas girada para a esquerda por um número

de bits depende do número da iteração. K_i é derivado desta chave girada por aplicando ainda outra transposição de 56 bits a ele. Um subconjunto diferente de 48 bits do 56 bits são extraídos e permutados em cada rodada.

Uma técnica que às vezes é usada para tornar o DES mais forte é chamada de **whitening**. Consiste em aplicar XOR a uma chave aleatória de 64 bits com cada bloco de texto simples antes alimentá-lo no DES e, em seguida, aplicar o XOR a uma segunda chave de 64 bits com o resultado texto cifrado antes de transmiti-lo. O clareamento pode ser facilmente removido executando o

Página 806

782
SEGURANÇA DE REDE
INDIVÍDUO. 8

operações reversas (se o receptor tiver as duas chaves de clareamento). Uma vez que esta técnica que efetivamente adiciona mais bits ao comprimento da chave, faz uma pesquisa exaustiva de o espaço-chave consome muito mais tempo. Observe que a mesma chave de clareamento é usado para cada bloco (ou seja, há apenas uma chave de clareamento).

DES foi envolvido em polêmica desde o dia em que foi lançado. isso foi baseado em uma cifra desenvolvida e patenteada pela IBM, chamada Lucifer, exceto que A cifra da IBM usava uma chave de 128 bits em vez de uma chave de 56 bits. Quando o US Federal O governo queria padronizar em uma cifra para uso não classificado, ele " convidou " IBM vai " discutir " o assunto com a NSA, a decifradora de códigos do governo dos EUA arm, que é o maior empregador mundial de matemáticos e criptologistas.

NSA é tão secreta que uma piada da indústria diz:

P: O que significa NSA?

R: Nenhuma agência.

Na verdade, NSA significa Agência de Segurança Nacional.

Após essas discussões, a IBM reduziu a chave de 128 bits para 56 bits e decidiu manter em segredo o processo pelo qual DES foi projetado. Muitos pessoas suspeitaram que o comprimento da chave foi reduzido para garantir que a NSA pudesse simplesmente quebrar o DES, mas nenhuma organização com um orçamento menor poderia. O ponto do o design secreto era supostamente para esconder uma porta dos fundos que poderia tornar ainda mais fácil para NSA quebrar DES. Quando um funcionário da NSA discretamente disse ao IEEE para cancelar um planejada conferência sobre criptografia, que não trouxe mais conforto às pessoas capaz. A NSA negou tudo.

Em 1977, dois pesquisadores de criptografia de Stanford, Diffie e Hellman (1977), projetou uma máquina para quebrar DES e estimou que poderia ser construída para 20 milhões dólares de leão. Dado um pequeno pedaço de texto simples e texto cifrado correspondente, este machine poderia encontrar a chave através da busca exaustiva do espaço da chave de 2^{56} entradas em 1 dia. Hoje em dia, o jogo acabou. Essa máquina existe, está à venda e custa menos de \$ 10.000 para fazer (Kumar et al., 2006).

DES triplo

Já em 1979, a IBM percebeu que o comprimento da chave DES era muito curto e visou uma maneira de aumentá-la efetivamente, usando criptografia tripla (Tuchman, 1979). O método escolhido, que desde então foi incorporado na International Standard 8732, é ilustrado na Fig. 8-8. Aqui, duas chaves e três estágios são usados. No primeiro estágio, o texto simples é criptografado usando DES da maneira usual com K_1 . No segundo estágio, o DES é executado no modo de descriptografia, usando K_2 como chave. Finalmente, outro

A criptografia DES é feita com K_1 .

Este desenho levanta imediatamente duas questões. Primeiro, por que são apenas dois chaves usadas, em vez de três? Em segundo lugar, porque é EDE (Encrypt Decrypt Encrypt) usado, em vez de EEE (Encrypt Encrypt Encrypt)? A razão de duas chaves serem usado é que mesmo o mais paranóico dos criptógrafos acredita que 112 bits é

Página 807

SEC. 8,2

ALGORITMOS DE CHAVE SIMÉTRICA

783

K₁

E

K₂

D

K₁

E

P

C

K₁

D

K₂

E

(uma)

(b)

K₁

D

C

P

Figura 8-8. (a) Criptografia tripla usando DES. (b) Descriptografia.

adequado para aplicações comerciais de rotina por enquanto. (E entre criptógrafos, a paranóia é considerada um recurso, não um bug.) Indo para 168 bits iria apenas adicionar a sobrecarga desnecessária de gerenciamento e transporte de outra chave para pouco ganho real.

O motivo para criptografar, descriptografar e criptografar novamente é o retrocesso compatibilidade com sistemas DES de chave única existentes. Tanto a criptografia quanto a de-

As funções de criptografia são mapeamentos entre conjuntos de números de 64 bits. De um cripto-ponto de vista gráfico, os dois mapeamentos são igualmente fortes. Usando EDE, como-sempre, em vez de EEE, um computador usando criptografia tripla pode falar com outro usando criptografia única apenas definindo $K_1 = K_2$. Esta propriedade permite criptografia tripla a ser implementado gradualmente, algo que não interessa aos criptógrafos acadêmicos mas de considerável importância para a IBM e seus clientes.

8.2.2 AES - O padrão de criptografia avançado

À medida que o DES começou a se aproximar do fim de sua vida útil, mesmo com DES triplo, NIST (Instituto Nacional de Padrões e Tecnologia), a agência dos EUA

Departamento de Comércio encarregado de aprovar padrões para o governo federal dos EUA governo, decidiu que o governo precisava de um novo padrão criptográfico para uso não classificado. O NIST estava ciente de toda a controvérsia em torno do DES e bem sabia que se acabasse de anunciar um novo padrão, todos sabendo coisa sobre criptografia assumiria automaticamente que a NSA havia construído um porta para que a NSA pudesse ler tudo criptografado com ele. Sob essas condições ções, provavelmente ninguém usaria o padrão e ele teria morrido silenciosamente.

Portanto, o NIST adotou uma abordagem surpreendentemente diferente para uma agência governamental

cracy: patrocinou um bake-off criptográfico (concurso). Em janeiro de 1997, re-pesquisadores de todo o mundo foram convidados a apresentar propostas para um novo padrão dard, a ser denominado **AES (Advanced Encryption Standard)**. As regras de cozimento estavam:

1. O algoritmo deve ser uma cifra de bloco simétrico.
2. O projeto completo deve ser público.
3. Os comprimentos de chave de 128, 192 e 256 bits devem ser suportados.

Página 808

784

SEGURANÇA DE REDE
INDIVÍDUO. 8

4. As implementações de software e hardware devem ser possíveis.

5. O algoritmo deve ser público ou licenciado em termos não discriminatórios.

Quinze propostas sérias foram feitas, e conferências públicas foram organizadas em que foram apresentados e os participantes foram ativamente incentivados a encontrar falhas em todos eles. Em agosto de 1998, o NIST selecionou cinco finalistas, principalmente com base de sua segurança, eficiência, simplicidade, flexibilidade e requisitos de memória (im-importante para sistemas embarcados). Mais conferências foram realizadas e mais fotos ocupado.

Em outubro de 2000, o NIST anunciou que havia selecionado Rijndael, de Joan Daehomens e Vincent Rijmen. O nome Rijndael, pronuncia-se boneca do Reno (mais ou menos), é derivado dos sobrenomes dos autores: Rijmen + Daemen. No

Novembro de 2001, Rijndael se tornou o padrão AES do governo dos EUA, publicado como FIPS (Federal Information Processing Standard) 197. Devido ao extraordinário abertura da competição, as propriedades técnicas de Rijndael, e o fato de que a equipe vencedora consistia em dois jovens criptógrafos belgas (que eram improvável de ter construído uma porta dos fundos apenas para agradar a NSA), Rijndael tornou-se o

cifra criptográfica dominante do mundo. A criptografia e descriptografia AES agora é parte do conjunto de instruções para alguns microprocessadores (por exemplo, Intel). Rijndael suporta comprimentos de chave e tamanhos de bloco de 128 bits a 256 bits em etapas de 32 bits. O comprimento da chave e o comprimento do bloco podem ser escolhidos independentemente.

No entanto, o AES especifica que o tamanho do bloco deve ser 128 bits e o comprimento da chave deve ser 128, 192 ou 256 bits. É duvidoso que alguém venha a usar 192 bits chaves, então, de fato, AES tem duas variantes: um bloco de 128 bits com uma chave de 128 bits e um

Bloco de 128 bits com chave de 256 bits.

Em nosso tratamento do algoritmo, examinaremos apenas o caso 128/128 porque isso provavelmente se tornará a norma comercial. Uma chave de 128 bits fornece uma chave espaço de $2^{128} \sim 3 \times 10^{38}$ teclas. Mesmo que a NSA consiga construir uma máquina com 1 bilhões de processadores paralelos, cada um podendo avaliar uma chave por picosegundo, levaria cerca de 10 a 10 anos para essa máquina pesquisar o espaço-chave. Até então o sol terá queimado, então as pessoas presentes terão que ler os resultados até luz de velas.

Rijndael

De uma perspectiva matemática, Rijndael é baseado na teoria de campo de Galois, o que lhe dá algumas propriedades de segurança comprováveis. No entanto, também pode ser visualizado como código C, sem entrar na matemática.

Como o DES, Rijndael usa substituição e permutações, e também usa múltiplas rodadas. O número de rodadas depende do tamanho da chave e do bloco, sendo 10 para chaves de 128 bits com blocos de 128 bits e até 14 para a maior chave ou o maior bloco. No entanto, ao contrário do DES, todas as operações envolvem bytes inteiros, para

Página 809

SEC. 8,2
ALGORITMOS DE CHAVE SIMÉTRICA

785

permitem implementações eficientes em hardware e software. Um esboço de o código é fornecido na Fig. 8-9. Observe que este código é para fins ilustrativos. Boas implementações de código de segurança seguirão práticas adicionais, como zerando a memória sensível após ela ter sido usada. Veja, por exemplo, Ferguson et al. (2010).

```
# define COMPRIMENTO 16
/* # bytes no bloco de dados ou chave */
#define NROWS 4
/* número de linhas no estado */
#define NCOLS 4
/* número de colunas no estado */
#define ROUNDS 10
/* número de iterações */
byte de caractere sem sinal de typedef;
/* inteiro sem sinal de 8 bits */
rijndael (texto simples de byte [LENGTH], texto cifrado de byte [LENGTH], chave de byte [LENGTH])
{
    int r;
    /* índice de loop */
    estado do byte [NROWS] [NCOLS];
    /* estado atual */
    struct {byte k [NROWS] [NCOLS];} rk [ROUNDS + 1]; /* teclas redondas */
    expandir chave (chave, rk);
    /* construir as teclas redondas */
    copiar texto simples para o estado (estado, texto simples);
    /* estado atual do init */
    xou tecla redonda no estado (estado, rk [0]);
    /* Chave XOR no estado */
    para (r = 1; r <= ROUNDS; r++) {
        substituto (estado);
        /* aplica S-box a cada byte */
        girar linhas (estado);
        /* girar a linha i por i bytes */
        if (r < ROUNDS) misturar colunas (estado);
        /* função de mistura */
        xou tecla redonda no estado (estado, rk [r]);
        /* Chave XOR no estado */
    }
}
```

```

estado de cópia para texto cifrado (texto cifrado, estado);
/* resultado de retorno */
}

```

Figura 8-9. Um esboço de Rijndael em C.

A função *rijndael* possui três parâmetros. São eles: *texto simples*, uma série de 16 bytes contendo os dados de entrada; *texto cifrado*, uma matriz de 16 bytes onde o codin a saída com phered será retornada; e *chave*, a *chave de 16 bytes*. Durante o cálculo, o estado atual dos dados é mantido em uma matriz de bytes, *estado*, cujo tamanho é $NROWS \times NCOLS$. Para blocos de 128 bits, essa matriz tem 4×4 bytes. Com 16 bytes, todo o bloco de dados de 128 bits pode ser armazenado.

A matriz de *estado* é inicializada para o texto simples e modificada em cada etapa do computação. Em algumas etapas, a substituição byte por byte é executada. Em outros, os bytes são permutados dentro do array. Outras transformações também são usadas. Em no final, o conteúdo do *estado* é retornado como o texto cifrado.

O código começa expandindo a chave em 11 matrizes do mesmo tamanho que o Estado. Eles são armazenados em *rk*, que é uma matriz de estruturas, cada uma contendo um estado array. Um deles será usado no início do cálculo e os outros 10 será usado durante as 10 rodadas, uma por rodada. O cálculo da rodada

Página 810

786

SEGURANÇA DE REDE INDIVÍDUO. 8

chaves da chave de criptografia é muito complicado para nós entrarmos aqui. Basta para dizer que as chaves redondas são produzidas por rotação repetida e XOR de variáveis grupos de bits-chave. Para todos os detalhes, consulte Daemen e Rijmen (2002).

A próxima etapa é copiar o texto simples para a matriz de *estado* para que possa ser processado durante as rodadas. É copiado em ordem de coluna, com os primeiros 4 bytes indo na coluna 0, os próximos 4 bytes indo para a coluna 1 e assim por diante. Ambas as colunas e as linhas são numeradas começando em 0, embora as rodadas sejam numeradas no início em 1. Esta configuração inicial das matrizes de 12 bytes de tamanho 4×4 é ilustrada em Fig. 8-10.

Estado	
rk [0]	
rk [1]	
rk [2]	
rk [3]	
rk [4]	
rk [5]	
rk [6]	
rk [7]	
rk [8]	
rk [9] rk [10]	

Texto simples de 128 bits
Chave de criptografia de 128 bits
Chaves redondas

Figura 8-10. Criação dos arrays *state* e *rk*.

Há mais uma etapa antes do início do cálculo principal: *rk* [0] é XORed no *estado*, byte por byte. Em outras palavras, cada um dos 16 bytes no *estado* é substituído pelo XOR de si mesmo e o byte correspondente em *rk* [0].

Agora é a hora da atração principal. O loop executa 10 iterações, uma por rodada, transformando o *estado* em cada iteração. O conteúdo de cada rodada é produzido em quatro etapas. A etapa 1 faz uma substituição de byte por byte no *estado*. Cada byte por sua vez, é usado como um índice em uma S-box para substituir seu valor pelo conteúdo daquele Entrada S-box. Esta etapa é uma cifra de substituição monoalfabética direta. Ao contrário DES, que tem várias S-box, Rijndael tem apenas uma S-box.

A etapa 2 gira cada uma das quatro linhas para a esquerda. A linha 0 é girada em 0 bytes (ou seja, não alterado), a linha 1 é girada 1 byte, a linha 2 é girada 2 bytes e a linha 3 é girada 3 bytes. Esta etapa difunde o conteúdo dos dados atuais em torno do bloco, analógico às permutações da Fig. 8-6.

O passo 3 mistura cada coluna independentemente das outras. A mistura é feito usando a multiplicação de matrizes em que a nova coluna é o produto do

coluna velha e uma matriz constante, com a multiplicação feita usando o finito Campo de Galois, $GF(2^8)$. Embora possa parecer complicado, existe um algoritmo que permite que cada elemento da nova coluna seja calculado usando duas tabelas ups e três XORs (Daemen e Rijmen, 2002, Apêndice E).

Página 811

SEC. 8,2

ALGORITMOS DE CHAVE SIMÉTRICA

787

Finalmente, a etapa 4 XORs a chave para esta rodada na matriz de *estado* para uso no proxima rodada.

Uma vez que cada etapa é reversível, a descriptografia pode ser feita apenas executando o algoritmo para trás. No entanto, há também um truque disponível em que a descriptografia pode ser feito executando o algoritmo de criptografia usando tabelas diferentes.

O algoritmo foi projetado não apenas para grande segurança, mas também para grande Rapidez. Uma boa implementação de software em uma máquina de 2 GHz deve ser capaz de atingir uma taxa de criptografia de 700 Mbps, que é rápida o suficiente para criptografar mais de 100 Vídeos MPEG-2 em tempo real. As implementações de hardware são ainda mais rápidas.

8.2.3 Modos de codificação

Apesar de toda essa complexidade, AES (ou DES, ou qualquer cifra de bloco para esse assunto) é basicamente uma cifra de substituição monoalfabética que usa caracteres grandes (128 bits caracteres para AES e caracteres de 64 bits para DES). Sempre que o mesmo texto simples bloco vai no front-end, o mesmo bloco de texto cifrado sai no back-end. E se você criptografa o texto simples *abcdefghijklm* 100 vezes com a mesma chave DES, você obtém o mesmo texto cifrado 100 vezes. Um intruso pode explorar esta propriedade para ajudar a subverter a cifra.

Modo de livro de código eletrônico

Para ver como esta propriedade de cifra de substituição monoalfabética pode ser usada para anular parcialmente a cifra, usaremos DES (triplo) porque é mais fácil de descrever Blocos de 64 bits do que blocos de 128 bits, mas o AES tem exatamente o mesmo problema. A maneira direta de usar DES para criptografar um longo trecho de texto simples é quebrá-lo em blocos consecutivos de 8 bytes (64 bits) e criptografá-los um após o outro com a mesma chave. A última parte do texto simples é preenchida com 64 bits, se necessário. Isto técnica é conhecida como **modo ECB** (**modo livro de códigos eletrônicos**) em analogia com livros de código antigos, onde cada palavra de texto simples era listada, seguida por seu texto cifrado (geralmente um número decimal de cinco dígitos).

Na Fig. 8-11, temos o início de um arquivo de computador listando os bônus anuais de empresa decidiu premiar a seus funcionários. Este arquivo consiste em consecutivas Registros de 32 bytes, um por funcionário, no formato mostrado: 16 bytes para o nome, 8 bytes para a posição e 8 bytes para o bônus. Cada um dos dezesseis 8 bytes blocos (numerados de 0 a 15) são criptografados por DES (triplo).

Leslie acabou de brigar com o chefe e não espera muito de um bônus.

Kim, ao contrário, é a favorita do chefe, e todo mundo sabe disso. Leslie pode obter ac-cessar o arquivo depois de criptografado, mas antes de ser enviado ao banco. Pode Leslie retificar esta situação injusta, dado apenas o arquivo criptografado?

Sem problema nenhum. Tudo o que Leslie precisa fazer é fazer uma cópia do 12º texto cifrado bloco (que contém o bônus de Kim) e use-o para substituir o quarto texto cifrado bloco (que contém o bônus de Leslie). Mesmo sem saber qual é o dia 12

Página 812

788

SEGURANÇA DE REDE
INDIVÍDUO. 8

Nome
Posição
Bônus
16
8

```

8
Bytes
Davis,
Obbie
Zelador
$
5
C ollins,
K im
Gerente
$ 1 0 0, 0 0 0
Preto ,
Robin
Patrão
$ 5 0 0, 0 0 0
A represas,
Leslie
Escriturário
$
1 0

```

Figura 8-11. O texto simples de um arquivo criptografado como 16 blocos DES. Block diz, Leslie pode esperar ter um Natal muito mais feliz este ano. (Cópia de-
o oitavo bloco de texto cifrado também é uma possibilidade, mas é mais provável que seja detectado
ed; além disso, Leslie não é uma pessoa gananciosa.)

Modo de encadeamento de blocos de criptografia

Para impedir esse tipo de ataque, todas as cifras de bloco podem ser encadeadas de várias maneiras de modo que substituir um bloco da maneira que Leslie fez fará com que o texto simples seja descriptografado

começando no bloco substituído para ser lixo. Uma forma de encadeamento é o **bloco de criptografia**

encadeamento. Neste método, mostrado na Fig. 8-12, cada bloco de texto simples é XORed com o bloco de texto cifrado anterior antes de ser criptografado. Conseqüentemente, o mesmo bloco de texto simples não mapeia mais no mesmo bloco de texto cifrado, e a criptografia não é mais uma grande cifra de substituição monoalfabética. O primeiro bloco é XORed com um **IV** escolhido aleatoriamente (**vetor de inicialização**), que é transmitido (de forma simples texto) junto com o texto cifrado.

```

(uma)
(b)
+
E
IV
Chave
Chave
IV
P 0
C 0
+
E
P 1
C 1
E
P 2
C 2
E
P 3
C 3
D
C 0
P 0
D
C 1
P 1
D
C 2
P 2
D
Decifrar
caixa
Encriptação
caixa
Exclusivo
OU
C 3
P 3
+
+

```

+

+

+

+

Figura 8-12. Encadeamento de blocos de cifras. (a) Criptografia. (b) Descriptografia.

Podemos ver como o modo de encadeamento de blocos de criptografia funciona examinando o exemplo

da Fig. 8-12. Começamos calculando $C_0 = E(P_0 \text{ XOR } IV)$. Então calculamos $C_1 = E(P_1 \text{ XOR } C_0)$, e assim por diante. A descriptografia também usa XOR para reverter o processo

ess, com $P_0 = IV \text{ XOR } D(C_0)$, e assim por diante. Observe que a criptografia do bloco i é um

Página 813

SEC. 8,2

ALGORITMOS DE CHAVE SIMÉTRICA

789

função de todo o texto simples nos blocos de 0 a $i - 1$, então o mesmo texto simples gera-
tes texto cifrado diferente dependendo de onde ocorre. Uma transformação do tipo Leslie feito resultará em absurdo por dois blocos começando com o bônus de Leslie campo. Para um oficial de segurança astuto, essa peculiaridade pode sugerir por onde começar a investigação que se seguiu.

O encadeamento de blocos de criptografia também tem a vantagem de que o mesmo bloco de texto simples

não resultará no mesmo bloco de texto cifrado, tornando a criptoanálise mais difícil.

Na verdade, esse é o principal motivo de seu uso.

Modo de feedback de cifra

No entanto, o encadeamento de blocos de cifras tem a desvantagem de exigir um inteiro Bloco de 64 bits para chegar antes que a descriptografia possa começar. Para criptografia byte a byte,

O modo de feedback de cifra usando DES (triplo) é usado, conforme mostrado na Fig. 8-13. Para AES, a ideia é exatamente a mesma, apenas um shift register de 128 bits é usado. Nesta figura, o estado da máquina de criptografia é mostrado depois que os bytes 0 a 9 têm sido criptografados e enviado. Quando o byte 10 de texto simples chega, conforme ilustrado na Fig. 8-13 (a), o algoritmo DES opera no registrador de deslocamento de 64 bits para gerar um registro de 64 bits

texto cifrado. O byte mais à esquerda desse texto cifrado é extraído e XORed com P_{10} .

Esse byte é transmitido na linha de transmissão. Além disso, o registrador de deslocamento é

deslocados 8 bits para a esquerda, fazendo com que C_2 caia da extremidade esquerda e C_{10} seja inserido no

posição apenas desocupada na extremidade direita por C_9 .

(uma)
Chave
 P_{10}
 C_{10}
 C_{10}
 C_{10}
E
Registrador de deslocamento de 64 bits
 $C_2 C_3 C_4 C_5 C_6 C_7 C_8 C_9$
Encriptação
caixa
Selecionar
byte mais à esquerda
Exclusivo ou
(b)
Chave
 C_{10}
 P_{10}
E
Registrador de deslocamento de 64 bits
 $C_2 C_3 C_4 C_5 C_6 C_7 C_8 C_9$

+

Encriptação
caixa
Selecionar

byte mais à esquerda

+

Figura 8-13. Modo de feedback de cifra. (a) Criptografia. (b) Descriptografia.

Observe que o conteúdo do registrador de deslocamento depende de todo o histórico anterior história do texto simples, então um padrão que se repete várias vezes no texto simples ser criptografado de maneira diferente a cada vez no texto cifrado. Tal como acontece com a cadeia de blocos de cifras

ing, um vetor de inicialização é necessário para começar a rolar a bola.

790

SEGURANÇA DE REDE INDIVÍDUO. 8

A descriptografia com modo de feedback de cifra funciona da mesma maneira que a criptografia. No particular, o conteúdo do shift register é *criptografado*, não *descriptografado*, de modo que o byte selecionado que é XORed com C_{10} para obter P_{10} é o mesmo que foi XORed com P_{10} para gerar C_{10} em primeiro lugar. Contanto que os dois registros de deslocamento permanecem idênticos, a descriptografia funciona corretamente. Isso é ilustrado na Fig. 8-13 (b). Um problema com o modo de feedback de cifra é que se um bit do texto cifrado for acidentalmente invertidos durante a transmissão, os 8 bytes que são descriptografados enquanto o byte incorreto no registrador de deslocamento será corrompido. Assim que o byte inválido for removido do registrador de deslocamento, o texto simples correto será gerado mais uma vez. Assim, o efeitos de um único bit invertido são relativamente localizados e não arruinam o resto do mensagem, mas eles estragam tantos bits quanto o registro de deslocamento é largo.

Modo de codificação de fluxo

No entanto, existem aplicativos em que ter um erro de transmissão de 1 bit bagunçar 64 bits de texto simples é um efeito muito grande. Para essas aplicações, um quarto opção, **modo de codificação de fluxo**, existe. Ele funciona criptografando uma inicialização vetor, usando uma chave para obter um bloco de saída. O bloco de saída é então criptografado, usando a chave para obter um segundo bloco de saída. Este bloco é então criptografado para obter um terceiro

bloco e assim por diante. A sequência (arbitrariamente grande) de blocos de saída, chamada de **keystream**, é tratado como um teclado de uso único e XORed com o texto simples para obter o texto cifrado, conforme mostrado na Fig. 8-14 (a). Observe que o IV é usado apenas no primeiro degrau. Depois disso, a saída é criptografada. Observe também que o keystream é independente dente dos dados, para que possam ser calculados com antecedência, se necessário, e são completamente

insensível a erros de transmissão. A descriptografia é mostrada na Fig. 8-14 (b).

E

(uma)

Chave

Texto simples

Texto cifrado

Keystream

Caixa de criptografia

IV

+

E

(b)

Chave

Texto simples

Texto cifrado

Keystream

Caixa de criptografia

IV

+

Figura 8-14. Uma cifra de fluxo. (a) Criptografia. (b) Descriptografia.

A descriptografia ocorre gerando o mesmo fluxo de chaves no lado receptor.

Uma vez que o keystream depende apenas do IV e da chave, ele não é afetado por erros de transmissão no texto cifrado. Assim, um erro de 1 bit na cifra transmitida o texto gera apenas um erro de 1 bit no texto simples descriptografado.

SEC. 8,2

ALGORITMOS DE CHAVE SIMÉTRICA

791

É essencial nunca usar o mesmo par (chave, IV) duas vezes com uma cifra de fluxo porque isso gerará o mesmo fluxo de chaves todas as vezes. Usando o mesmo keystream expõe duas vezes o texto cifrado a um **ataque de reutilização de keystream**. Imagine que o bloco de texto simples, P_0 , é criptografado com o fluxo de chaves para obter $P_0 \text{ XOR } K_0$. Mais tarde, um segundo bloco de texto simples, Q_0 , é criptografado com o mesmo fluxo de chave para obter

$Q_0 \text{ XOR } K_0$. Um intruso que captura ambos os blocos de texto cifrado pode sim- Use XOR para obter $P_0 \text{ XOR } Q_0$, que elimina a chave. O intruso er agora tem o XOR dos dois blocos de texto simples. Se um deles é conhecido ou pode ser adivinhado, o outro também pode ser encontrado. Em qualquer caso, o XOR de dois textos simples

streams podem ser atacados usando propriedades estatísticas da mensagem. Para exemplo, para texto em inglês, o caractere mais comum no fluxo provavelmente será o XOR de dois espaços, seguido do XOR de espaço e a letra "e", etc.

resumindo, equipado com o XOR de dois textos simples, o criptanalista tem um excelente chance de deduzir ambos.

Modo de contador

Um problema que todos os modos, exceto o modo de livro de código eletrônico, têm é que o acesso aleatório aos dados criptografados é impossível. Por exemplo, suponha que um arquivo seja transmitido por uma rede e armazenado em disco de forma criptografada. Isso pode ser uma maneira razoável de operar se o computador receptor for um notebook que pode ser roubado. Armazenar todos os arquivos críticos em forma criptografada reduz bastante o dano devido ao vazamento de informações secretas no caso de o computador cai nas mãos erradas.

No entanto, os arquivos de disco são frequentemente acessados em ordem não sequencial, especialmente os arquivos

em bancos de dados. Com um arquivo criptografado usando encadeamento de blocos de criptografia, acessar um arquivo

bloco dom requer primeiro descriptografar todos os blocos à frente dele, um caro proposta. Por esta razão, ainda outro modo foi inventado: **modo contador**, conforme ilustrado na Fig. 8-15. Aqui, o texto simples não é criptografado diretamente. Em vez de, o vetor de inicialização mais uma constante é criptografado, e o texto cifrado resultante é XORed com o texto simples. Ao incrementar o vetor de inicialização em 1 para cada novo bloco, é fácil descriptografar um bloco em qualquer lugar do arquivo sem ter que primeiro descompactar

criptografar todos os seus predecessores.

Embora o modo contador seja útil, ele tem uma fraqueza que vale a pena destacar.

Suponha que a mesma chave, K , seja usada novamente no futuro (com um plano diferente texto, mas o mesmo IV) e um invasor adquire todo o texto cifrado de ambas as execuções.

Os fluxos de chaves são os mesmos em ambos os casos, expondo a cifra a um fluxo de chaves reutilize o ataque do mesmo tipo que vimos com cifras de fluxo. Todo o criptanalista tem que fazer é XOR os dois textos cifrados juntos para eliminar todos os criptográficos proteção e é só pegar o XOR dos textos simples. Essa fraqueza não significa

o modo contador é uma má ideia. Significa apenas que tanto as chaves quanto os vetores de inicialização

devem ser escolhidos de forma independente e aleatória. Mesmo se a mesma chave for acidentalmente usado duas vezes, se o IV for diferente a cada vez, o texto simples está seguro.

792

SEGURANÇA DE REDE

INDIVÍDUO. 8

Encriptação

```

caixa
+
E
IV
Chave
P0
C0
+
E
IV + 1
Chave
P1
C1
+
E
IV + 2
Chave
P2
C2
+
E
IV + 3
Chave
P3
C3

```

Figura 8-15. Criptografia usando o modo de contador.

8.2.4 Outras Cifras

AES (Rijndael) e DES são os algoritmos criptográficos de chave simétrica mais conhecidos. (Ninguém vai culpar você se você usar AES em seu produto e o AES estiver quebrado, mas eles vão certamente culpar você se você usar uma cifra fora do padrão e depois for quebrada.) Como-nunca, vale a pena mencionar que várias outras cifras de chave simétrica foram planejado. Alguns deles estão embutidos em vários produtos. Um pouco mais os comuns estão listados na Fig. 8-16. É possível usar combinações destes cifras, por exemplo, AES sobre Twofish, de modo que ambas as cifras precisam ser quebradas para recuperar os dados.

Cifra

Autor

Comprimento da chave

Comentários

DES

IBM

56 bits

Muito fraco para usar agora

RC4

Ronald Rivest

1–2048 bits

Cuidado: algumas teclas são fracas

RC5

Ronald Rivest

128–256 bits

Bom, mas patenteado

AES (Rijndael)

Daemen e Rijmen

128–256 bits

Melhor escolha

Serpente

Anderson, Biham, Knudsen

128–256 bits

Muito forte

DES triplo

IBM

168 bits

Bom, mas ficando velho

Dois peixes

Bruce Schneier

128–256 bits

Muito forte; amplamente utilizado

Figura 8-16. Alguns algoritmos criptográficos de chave simétrica comuns.

8.2.5 Criptoanálise

Antes de deixar o assunto da criptografia de chave simétrica, vale a pena pelo menos mencionando quatro desenvolvimentos em criptanálise. O primeiro desenvolvimento é **diferente criptoanálise ferencial** (Biham e Shamir, 1997). Esta técnica pode ser usada

Página 817

SEC. 8,2

ALGORITMOS DE CHAVE SIMÉTRICA

793

para atacar qualquer cifra de bloco. Funciona começando com um par de blocos de texto simples diferindo apenas em um pequeno número de bits e observando cuidadosamente o que acontece em cada iteração interna à medida que a criptografia prossegue. Em muitos casos, alguns bits andorinhas são mais comuns do que outras, o que pode levar a ataques probabilísticos.

O segundo desenvolvimento digno de nota é **a criptoanálise linear** (Matsui, 1994).

Ele pode quebrar o DES com apenas 2⁴³ textos simples conhecidos. Funciona por XORing certo bits no texto simples e no texto cifrado juntos e examinando o resultado. Quando terminar repetidamente, metade dos bits deve ser 0s e a outra metade deve ser 1s. Freqüentemente, entretanto, cifras introduzem um viés em uma direção ou outra, e esse viés, embora pequeno, pode ser explorado para reduzir o fator de trabalho. Para mais detalhes, consulte o artigo de Matsui. O terceiro desenvolvimento está usando a análise do consumo de energia elétrica para encontrar chaves secretas. Os computadores normalmente usam cerca de 3 volts para representar 1 bit e 0

volts para representar um bit 0. Assim, o processamento de 1 consome mais energia elétrica do que processando um 0. Se um algoritmo criptográfico consiste em um loop no qual a chave bits são processados em ordem, um invasor que substitui o relógio principal de n - GHz por um relógio lento (por exemplo, 100 Hz) e coloca garras de crocodilo na alimentação e no aterramento da CPU

os pinos podem monitorar com precisão a energia consumida por cada instrução da máquina. A partir desses dados, deduzir a chave é surpreendentemente fácil. Este tipo de criptoanálise pode ser derrotado apenas codificando cuidadosamente o algoritmo em linguagem assembly para certifique-se de que o consumo de energia seja independente da chave e também independente de todas as teclas redondas individuais.

O quarto desenvolvimento é a análise de tempo. Os algoritmos criptográficos estão cheios de instruções if que testam bits nas chaves redondas. Se as partes then e else forem diferentes diferentes quantidades de tempo, desacelerando o relógio e vendo quanto tempo etapas executadas, também pode ser possível deduzir as chaves redondas. De uma vez chaves são conhecidas, a chave original geralmente pode ser calculada. Poder e tempo a análise também pode ser empregada simultaneamente para tornar o trabalho mais fácil. Enquanto pow-

er e análise de tempo podem parecer exóticas, na realidade são técnicas poderosas que pode quebrar qualquer cifra não projetada especificamente para resistir a eles.

8.3 ALGORITMOS DE CHAVE PÚBLICA

Historicamente, distribuir as chaves sempre foi o elo mais fraco na maioria criptosistemas. Não importa o quanto forte seja um criptosistema, se um intruso pudesse roubar a chave, o sistema não valia a pena. Os criptologistas sempre deram como certo que a chave de criptografia e a chave de descriptografia eram as mesmas (ou facilmente derivadas de um outro). Mas a chave precisava ser distribuída a todos os usuários do sistema. Assim, parecia que havia um problema inerente. As chaves deveriam ser protegidas contra roubo, mas também tinham de ser distribuídos, para que não pudessem ser trancados no cofre de um banco. Em 1976, dois pesquisadores da Universidade de Stanford, Diffie e Hellman (1976), propôs um tipo radicalmente novo de criptosistema, no qual a criptografia e as chaves de descriptografia eram tão diferentes que a chave de descriptografia não poderia ser

Página 818

794

SEGURANÇA DE REDE

INDIVÍDUO. 8

derivado da chave de criptografia. Em sua proposta, o algoritmo de criptografia (com chave) ritmo, E , e o algoritmo de descriptografia (com chave), D , tinham que atender a três requisitos. Esses requisitos podem ser definidos simplesmente da seguinte forma:

1. $D(E(P)) = P$.

2. É extremamente difícil deduzir D a partir de E .

3. E não pode ser quebrado por um ataque de texto simples escolhido.

O primeiro requisito diz que se aplicarmos D a uma mensagem criptografada, $E(P)$, nós obter a mensagem de texto simples original, P , de volta. Sem esta propriedade, o legítimo o receptor não conseguiu decifrar o texto cifrado. O segundo requisito fala por ele-
auto. O terceiro requisito é necessário porque, como veremos em um momento,
truders podem experimentar o algoritmo o quanto quiserem. Sob estes
condições, não há razão para que a chave de criptografia não possa ser tornada pública.
O método funciona assim. Uma pessoa, digamos, Alice, que deseja receber segredo
mensagens, primeiro concebe dois algoritmos que atendem aos requisitos acima. O en-
algoritmo de criptografia e a chave de Alice são, então, tornados públicos, daí o nome **publico-
criptografia de chave**. Alice pode colocar sua chave pública em sua página inicial no
Web, por exemplo. Usaremos a notação E_A para significar o algoritmo de criptografia
parametrizado pela chave pública de Alice. Da mesma forma, o algoritmo de descriptografia
(segredo)

parametrizado pela chave privada de Alice é D_A . Bob faz a mesma coisa, divulgando
 E_B , mas mantendo D_B em segredo.

Agora vamos ver se podemos resolver o problema de estabelecer um canal seguro
entre Alice e Bob, que nunca tiveram nenhum contato anterior. De Alice
chave de criptografia, E_A , e a chave de criptografia de Bob, E_B , são consideradas publicamente
arquivos legíveis. Agora Alice pega sua primeira mensagem, P , calcula $E_B(P)$, e envia
para Bob. Bob então o descriptografa aplicando sua chave secreta D_B [ou seja, ele calcula
 $D_B(E_B(P)) = P$]. Ninguém mais pode ler a mensagem criptografada, $E_B(P)$, porque
o sistema de criptografia é considerado forte e porque é muito difícil
derivam D_B a partir do conhecido publicamente E_B . Para enviar uma resposta, R , Bob
transmite $E_A(R)$.

Alice e Bob agora podem se comunicar com segurança.

Uma nota sobre a terminologia talvez seja útil aqui. Criptografia de chave pública re-
exige que cada usuário tenha duas chaves: uma chave pública, usada por todo o mundo para en-
criptografar mensagens a serem enviadas a esse usuário e uma chave privada, de que o usuário
precisa
para descriptografar mensagens. Iremos consistentemente nos referir a essas chaves como o *público*
e chaves *privadas*, respectivamente, e as distingue das chaves *secretas* usadas para
criptografia de chave simétrica convencional.

8.3.1 RSA

O único problema é que precisamos encontrar algoritmos que realmente satisfaçam todos os três
requisitos. Devido às vantagens potenciais da criptografia de chave pública, muitos
os pesquisadores estão trabalhando arduamente e alguns algoritmos já foram publicados.

Página 819

SEC. 8,3

ALGORITMOS DE CHAVE PÚBLICA

795

Um bom método foi descoberto por um grupo do MIT (Rivest et al., 1978). Isto é
conhecido pelas iniciais dos três descobridores (Rivest, Shamir, Adleman): **RSA**. isto
sobreviveu a todas as tentativas de quebrá-lo por mais de 30 anos e é considerado muito
Forte. Muita segurança prática se baseia nele. Por esta razão, Rivest, Shamir,
e Adleman receberam o prêmio ACM Turing de 2002. Sua maior desvantagem é
que requer chaves de pelo menos 1024 bits para uma boa segurança (contra 128 bits para
algoritmos de chave simétrica), o que o torna bastante lento.

O método RSA é baseado em alguns princípios da teoria dos números. Nós vamos
agora resumir como usar o método; para detalhes, consulte o jornal.

1. Escolha dois números primos grandes, p e q (tipicamente 1024 bits).
2. Calcule $n = p \times q$ e $z = (p - 1) \times (q - 1)$.
3. Escolha um número relativamente primo a z e chame-o de d .
4. Encontre e tal que $e \times d = 1 \text{ mod } z$.

Com esses parâmetros calculados com antecedência, estamos prontos para iniciar a criptografia. Divida o texto simples (considerado como uma string de bits) em blocos, de modo que cada texto simples

a mensagem, P , caia no intervalo $0 \leq P < n$. Faça isso agrupando o texto simples em blocos de k bits, onde k é o maior inteiro para o qual $2^k < n$ é verdadeiro.

Para criptografar uma mensagem, P , calcule $C = P^e \pmod{n}$. Para descriptografar C , calcule $P = C^d \pmod{n}$. Pode ser provado que para todo P na faixa especificada, o en- as funções de criptografia e descriptografia são inversas. Para realizar a criptografia, você precisa e e n . Para realizar a descriptografia, você precisa de d e n . Portanto, o público a chave consiste no par (e, n) e a chave privada consiste em (d, n) .

A segurança do método é baseada na dificuldade de fatorar um grande número bers. Se o criptanalista pudesse fatorar o (publicamente conhecido) n , ele poderia encontrar p e q , e destes z . Equipado com o conhecimento de z e e , d pode ser encontrado usando o algoritmo de Euclides. Felizmente, os matemáticos têm tentado fatorar grandes números por pelo menos 300 anos, e as evidências acumuladas sugerem que é um problema extremamente difícil.

De acordo com Rivest e colegas, fatorar um número de 500 dígitos re- quire 10²⁵ anos usando força bruta. Em ambos os casos, eles assumiram o mais conhecido algoritmo e um computador com tempo de instrução de 1 μseg. Com um milhão de fichas rodando ning em paralelo, cada um com um tempo de instrução de 1 nseg, ainda levaria 10¹⁶ anos. Mesmo que os computadores continuem a ficar mais rápidos em uma ordem de magnitude por década, levará muitos anos antes que a fatoração de um número de 500 dígitos se torne viável ble, momento em que os nossos descendentes pode simplesmente escolher p e q ainda maior.

Um exemplo pedagógico trivial de como o algoritmo RSA funciona é dado em

Fig. 8-17. Para este exemplo, escolhemos $p = 3$ e $q = 11$, dando $n = 33$ e $z = 20$. Um valor adequado para d é $d = 7$, uma vez que 7 e 20 não têm fatores comuns.

Com essas escolhas, e pode ser encontrado resolvendo a equação $7^e \equiv 1 \pmod{20}$, que produz $e = 3$. O texto cifrado, C , correspondendo a uma mensagem de texto simples, P , é

Página 820

796

SEGURANÇA DE REDE INDIVÍDUO. 8

dado por $C = P^e \pmod{33}$. O texto cifrado é descriptografado pelo receptor por mak- uso da regra $P = C^d \pmod{33}$. A figura mostra a criptografia do texto simples "SUZANNE" como um exemplo.

Simbólico

S

você

Z

UMA

N

N

E

Simbólico

S

você

Z

UMA

N

N

E

Numérico

Texto simples (P)

Texto cifrado (C)

Após a descriptografia

Cálculo do receptor

Cálculo do remetente

19

21

26

01

```

14
14
05
19
21
26
01
14
14
05
P3
6859
9261
17576
1
2744
2744
125
P3 (mod 33)
C7 (mod 33)
28
21
20
1
5
5
26
C7
13492928512
1801088541
1280000000
1
78125
78125
8031810176

```

Figura 8-17. Um exemplo do algoritmo RSA.

Como os primos escolhidos para este exemplo são tão pequenos, P deve ser menor que 33, então cada bloco de texto simples pode conter apenas um único caractere. O resultado é um cifra de substituição monoalfabética, não muito impressionante. Se, em vez disso, tivéssemos sen p e $q \sim 2^{512}$, teríamos $n \sim 2^{1024}$, de modo que cada bloco poderia ser até 1024 bits ou 128 caracteres de oito bits, contra 8 caracteres para DES e 16 caracteres para AES.

Deve ser apontado que usar RSA como descrevemos é semelhante a usando um algoritmo simétrico no modo ECB - o mesmo bloco de entrada dá o mesmo bloco de saída. Portanto, alguma forma de encadeamento é necessária para a criptografia de dados. No entanto, na prática, a maioria dos sistemas baseados em RSA usam criptografia de chave pública pri-marily para distribuir chaves de sessão única para uso com algumas chaves simétricas algoritmo como AES ou DES triplo. RSA é muito lento para criptografar grandes volumes de dados, mas é amplamente utilizado para distribuição de chaves.

8.3.2 Outros Algoritmos de Chave Pública

Embora RSA seja amplamente usado, não é de forma alguma o único algoritmo de chave pública conhecido. O primeiro algoritmo de chave pública foi o algoritmo da mochila (Merkle e Hellman, 1978). A ideia aqui é que alguém possui um grande número de objetos, cada um com um peso diferente. O proprietário codifica a mensagem secretamente selecionando-carregar um subconjunto de objetos e colocá-los na mochila. O peso total de os objetos na mochila são tornados públicos, assim como a lista de todos os objetos possíveis e seus pesos correspondentes. A lista de objetos na mochila é mantida em segredo. Com certas restrições adicionais, o problema de descobrir uma possível lista de objetos com o peso dado eram considerados computacionalmente inviáveis e formou a base do algoritmo de chave pública.

Adi Shamir (o "S" na RSA) prontamente quebrou e recebeu a recompensa. Implacável, Merkle fortaleceu o algoritmo e ofereceu uma recompensa de \$ 1000 para qualquer um que pudesse quebrar o novo. Ronald Rivest (o "R" na RSA) prontamente quebrou o novo e recebeu a recompensa. Merkle não se atreveu a oferecer \$ 10.000 para a próxima versão, então "A" (Leonard Adleman) estava sem sorte. Mesmo assim, o algoritmo da mochila não é considerado seguro e não é usado na prática por nenhum Mais.

Outros esquemas de chave pública são baseados na dificuldade de computação discreta logaritmos. Algoritmos que usam este princípio foram inventados por El Gamal (1985) e Schnorr (1991).

Existem alguns outros esquemas, como aqueles baseados em curvas elípticas (Menezes e Vanstone, 1993), mas as duas categorias principais são aquelas baseadas no diffi-culdade de fatorar grandes números e calcular logaritmos discretos módulo a grande primo. Esses problemas são considerados genuinamente difíceis de resolver - matemáticos têm trabalhado neles por muitos anos, sem qualquer grande descobertas.

8.4 ASSINATURAS DIGITAIS

A autenticidade de muitos documentos legais, financeiros e outros é determinada pela presença ou ausência de assinatura manuscrita autorizada. E photocópias não contam. Para sistemas de mensagens computadorizados para substituir o transporte físico de documentos em papel e tinta, um método deve ser encontrado para permitir documentos a serem assinados de forma imprevisível.

O problema de conceber uma substituição para assinaturas manuscritas é uma dificuldade culto um. Basicamente, o que é necessário é um sistema pelo qual uma das partes possa enviar um mensagem assinada para outra parte de forma que as seguintes condições sejam válidas:

1. O destinatário pode verificar a identidade reivindicada do remetente.
2. O remetente não pode repudiar posteriormente o conteúdo da mensagem.
3. O receptor não pode ter inventado a mensagem ele mesmo.

O primeiro requisito é necessário, por exemplo, em sistemas financeiros. Quando um computador do cliente pede ao computador de um banco para comprar uma tonelada de ouro, o computador precisa ser capaz de se certificar de que o computador que está dando o pedido realmente

pertence ao cliente cuja conta será debitada. Em outras palavras, o banco precisa autenticar o cliente (e o cliente precisa autenticar o banco).

O segundo requisito é necessário para proteger o banco contra fraudes. Suponha que o banco compra a tonelada de ouro, e imediatamente depois o preço do ouro

Página 822

798

SEGURANÇA DE REDE
INDIVÍDUO. 8

cai drasticamente. Um cliente desonesto pode então processar o banco, alegando que ele nunca emitiu qualquer ordem para comprar ouro. Quando o banco produz a mensagem no tribunal, o cliente pode negar o envio. A propriedade que não faz parte de um contrato pode negar posteriormente tê-lo assinado é chamado de não- **repúdio**. O sinal digital os esquemas da natureza que estudaremos agora ajudam a fornecê-lo.

O terceiro requisito é necessário para proteger o cliente no caso de preço do ouro dispara e o banco tenta construir uma mensagem assinada em que o cliente pediu uma barra de ouro em vez de uma tonelada. Neste cenário de fraude, o banco apenas fica com o resto do ouro para si.

8.4.1 Assinaturas de chave simétrica

Uma abordagem para assinaturas digitais é ter uma autoridade central que conhece tudo e em quem todos confiam, digamos, Big Brother (BB). Cada usuário então escolhe uma chave secreta e a leva à mão para o escritório de BB. Assim, apenas Alice e BB sabe a chave secreta de Alice, K_A e assim por diante.

Quando Alice deseja enviar uma mensagem assinada em texto simples, P , para seu banqueiro, Bob, ela gera $K_A(B, R_A, t, P)$, onde B é a identidade de Bob, R_A é um número aleatório

escolhido por Alice, t é um carimbo de data / hora para garantir o frescor e $K_A(B, R_A, t, P)$ é o mensageiro criptografado com sua chave, K_A . Em seguida, ela o envia conforme ilustrado na Figura 8-18.

BB vê que a mensagem é de Alice, descriptografa e envia uma mensagem para Bob como mostrando. A mensagem para Bob contém o texto simples da mensagem de Alice e também a mensagem assinada $K_{BB}(A, t, P)$. Bob agora atende ao pedido de Alice.

A, $K_A(B, R_A, t, P)$
Prumo
Alice
BB
 $K_B(A, R_A, t, P, K_{BB}(A, t, P))$
1
2

Figura 8-18. Assinaturas digitais com Big Brother.

O que acontecerá se Alice negar posteriormente o envio da mensagem? A etapa 1 é que todos um processa todo mundo (pelo menos, nos Estados Unidos). Finalmente, quando o caso chegar a tribunal e Alice vigorosamente nega ter enviado a Bob a mensagem contestada, o juiz perguntará a Bob como ele pode ter certeza de que a mensagem contestada veio de Alice e não de Trudy. Bob primeiro aponta que BB não aceitará uma mensagem de Alice a menos que seja criptografado com K_A , então não há possibilidade de Trudy enviar BB a mensagem falsa de Alice sem BB detectá-la imediatamente.

Bob então produz dramaticamente a Figura A: $K_{BB}(A, t, P)$. Bob diz que este é uma mensagem assinada por BB que prova que Alice enviou P para Bob. O juiz então pergunta a BB (em quem todos confiam) para descriptografar o Anexo A. Quando BB testemunha que Bob é tel-

Na verdade, o juiz decide a favor de Bob. Caso arquivado.

Página 823

SEC. 8,4

ASSINATURAS DIGITAIS

799

Um problema potencial com o protocolo de assinatura da Fig. 8-18 é Trudy re-jogando qualquer mensagem. Para minimizar esse problema, carimbos de data / hora são usados por meio de

Fora. Além disso, Bob pode verificar todas as mensagens recentes para ver se R_A foi usado em qualquer

deles. Nesse caso, a mensagem é descartada como uma repetição. Observe que com base no tempo-carimbo, Bob rejeitará mensagens muito antigas. Para se proteger contra ataques de repetição instantânea,

Bob apenas verifica o R_A de cada mensagem recebida para ver se tal mensagem recebido de Alice na última hora. Caso contrário, Bob pode assumir com segurança que este é um novo pedido.

8.4.2 Assinaturas de chave pública

Um problema estrutural com o uso de criptografia de chave simétrica para sinal digital naturezas é que todos têm que concordar em confiar no Big Brother. Além disso, grande O irmão consegue ler todas as mensagens assinadas. Os candidatos mais lógicos para concorrer o servidor do Big Brother são o governo, os bancos, os contadores e os advogados. Infelizmente, nada disso inspira total confiança em todos os cidadãos.

Portanto, seria bom se a assinatura de documentos não exigisse uma autoridade confiável.

Felizmente, a criptografia de chave pública pode dar uma contribuição importante na esta área. Vamos supor que os algoritmos de criptografia e descriptografia de chave pública tem a propriedade de que $E(D(P)) = P$, além, é claro, da propriedade usual que $D(E(P)) = P$. (RSA tem essa propriedade, então a suposição não é irreal confiável.) Supondo que seja esse o caso, Alice pode enviar uma mensagem de texto simples assinada

sábio, P , para Bob, transmitindo $E_B(D_A(P))$. Observe com atenção que Alice a conhece própria chave (privada), D_A , bem como a chave pública de Bob, E_B , portanto, construir esta mensagem

sage é algo que Alice pode fazer.

Quando Bob recebe a mensagem, ele a transforma usando sua chave privada, como

usual, rendendo $D_A(P)$, como mostrado na Fig. 8-19. Ele armazena este texto em um lugar seguro e então aplica E_A para obter o texto simples original.

```
Bob's
chave pública,
E_B
Alice's
chave privada,
D_A
Bob's
chave privada,
D_B
D_A(P)
D_A(P)
E_B(D_A(P))
Linha de transmissão
Computador da Alice
Computador de Bob
P
P
Alice's
chave pública,
E_A
```

Figura 8-19. Assinaturas digitais usando criptografia de chave pública.

Para ver como a propriedade de assinatura funciona, suponha que Alice posteriormente nega ter enviado a mensagem P a Bob. Quando o caso chega ao tribunal, Bob pode produzir P e $D_A(P)$. O juiz pode facilmente verificar se Bob realmente tem um mensagem válida criptografada por D_A simplesmente aplicando E_A a ela. Já que Bob não

Página 824

800

SEGURANÇA DE REDE INDIVÍDUO. 8

sabe o que é a chave privada de Alice, a única maneira de Bob ter adquirido uma mensagem sage criptografado por ele é se Alice realmente o enviou. Enquanto na prisão por perjúrio e fraude, Alice terá muito tempo para desenvolver novos algoritmos de chave pública interessantes. Embora usar criptografia de chave pública para assinaturas digitais seja um elegante esquema, existem problemas que estão relacionados ao ambiente em que operam comeu em vez do algoritmo básico. Por um lado, Bob pode provar que uma mensagem sage foi enviado por Alice apenas enquanto D_A permanecer em segredo. Se Alice a revelar chave secreta, o argumento não é mais válido, porque qualquer um poderia ter enviado o mensagem, incluindo o próprio Bob.

O problema pode surgir, por exemplo, se Bob for o corretor da bolsa de Alice. Suponha que Alice diz a Bob para comprar uma determinada ação ou título. Imediatamente depois disso, o preço cai drasticamente. Para repudiar sua mensagem para Bob, Alice corre para a polícia alegando que sua casa foi assaltada e o PC com sua chave foi roubado.

Dependendo das leis de seu estado ou país, ela pode ou não ser legalmente responsável, especialmente se ela alegar não ter descoberto a invasão até obter do trabalho, várias horas depois do suposto acontecimento.

Outro problema com o esquema de assinatura é o que acontece se Alice decidir para mudar sua chave. Fazer isso é claramente legal e provavelmente é uma boa ideia fazer tão periodicamente. Se um processo judicial surgir posteriormente, conforme descrito acima, o juiz irá

aplicar a corrente E_{UM} para $D_A(P)$ e que descubra que não produz P . Bob vai parecer muito estúpido neste ponto.

Em princípio, qualquer algoritmo de chave pública pode ser usado para assinaturas digitais. O padrão de fato da indústria é o algoritmo RSA. Muitos produtos de segurança o utilizam. No entanto, em 1991, o NIST propôs o uso de uma variante da chave pública El Gamal algoritm para seu novo **padrão de assinatura digital (DSS)**. El Gamal obtém seu seguro da dificuldade de calcular logaritmos discretos, ao invés da dificuldade culdade de fatorar grandes números.

Como de costume, quando o governo tenta ditar padrões criptográficos, há foi um alvoroço. DSS foi criticado por ser

1. Muito secreto (NSA elaborou o protocolo para usar El Gamal).
2. Muito lento (10 a 40 vezes mais lento do que RSA para verificar assinaturas).

3. Muito novo (El Gamal ainda não tinha sido analisado minuciosamente).

4. Muito inseguro (chave fixa de 512 bits).

Em uma revisão subsequente, o quarto ponto foi considerado discutível quando as chaves até 1.024 bits foram permitidos. No entanto, os primeiros dois pontos permanecem válidos.

8.4.3 Resumos de mensagens

Uma crítica aos métodos de assinatura é que eles costumam acoplar duas funções distintas: autenticação e sigilo. Muitas vezes, a autenticação é necessária, mas o sigilo é nem sempre necessário. Além disso, obter uma licença de exportação é geralmente mais fácil se o sistema em

Página 825

SEC. 8,4

ASSINATURAS DIGITAIS

801

question fornece apenas autenticação, mas não sigilo. Abaixo, descreveremos um esquema de autenticação que não requer criptografar a mensagem inteira.

Este esquema é baseado na ideia de uma função hash unilateral que leva um pedaço de texto simples arbitrariamente longo e a partir dele calcula uma string de bits de comprimento fixo.

Esta função hash, MD , muitas vezes chamada de **resumo de mensagem**, tem quatro propriedades importantes

laços:

1. Dado P , é fácil calcular $MD(P)$.
2. Dado $MD(P)$, é efetivamente impossível encontrar P .
3. Dado P , ninguém pode encontrar P' tal que $MD(P') = MD(P)$.
4. Uma mudança na entrada de até 1 bit produz uma saída muito diferente.

Para atender ao critério 3, o hash deve ter pelo menos 128 bits, de preferência mais. Para atender ao critério 4, o hash deve destruir os bits muito bem, não muito diferente do algoritmos de criptografia de chave simétrica que vimos.

Calcular um resumo de mensagem de um pedaço de texto simples é muito mais rápido do que encriptografar esse texto simples com um algoritmo de chave pública, para que as mensagens resumidas possam ser

usado para acelerar algoritmos de assinatura digital. Para ver como isso funciona, considere o protocolo de assinatura da Figura 8-18 novamente. Em vez de assinar P com $K_{BB}(A, t, P)$, BB agora calcula o resumo da mensagem aplicando MD a P , gerando $MD(P)$. BB em seguida, inclui $K_{BB}(A, t, MD(P))$ como o quinto item na lista criptografada com K_B que é enviado para Bob, em vez de $K_{BB}(A, t, P)$.

Se surgir uma disputa, Bob pode produzir P e $K_{BB}(A, t, MD(P))$. After Big O irmão o descriptografou para o juiz, Bob tem $MD(P)$, que é garantido genuíno, e a alegada P . No entanto, uma vez que é efetivamente impossível para Bob encontrar qualquer outra mensagem que dê esse hash, o juiz ficará facilmente convencido de que Bob está dizendo a verdade. Usar resumos de mensagens dessa forma economiza criptografia custos de tempo e transporte de mensagens.

Os resumos de mensagens também funcionam em sistemas criptográficos de chave pública, conforme mostrado na Fig. 8-

20. Aqui, Alice primeiro calcula o resumo da mensagem de seu texto simples. Ela então assina o resumo da mensagem e envia o resumo assinado e o texto simples para Bob. E se Trudy substitui P ao longo do caminho, Bob verá isso quando calcular $MD(P)$.

P, $D_A(MD(P))$

Prumo

Alice

Figura 8-20. Assinaturas digitais usando resumos de mensagens.

Página 826

802

SEGURANÇA DE REDE

INDIVÍDUO. 8

SHA-1 e SHA-2

Uma variedade de funções de resumo da mensagem foram propostas. Um dos mais A função amplamente utilizada é **SHA-1 (Secure Hash Algorithm 1)** (NIST, 1993). Gostar todos os resumos de mensagens, ele opera mutilando bits de uma forma suficientemente complicada que cada bit de saída é afetado por cada bit de entrada. SHA-1 foi desenvolvido pela NSA e abençoado pelo NIST no FIPS 180-1. Ele processa dados de entrada em blocos de 512 bits e ele gera um resumo da mensagem de 160 bits. Uma maneira típica de Alice enviar uma mensagem não secreta

mas a mensagem assinada para Bob é ilustrada na Figura 8-21. Aqui, sua mensagem de texto simples é alimentado no algoritmo SHA-1 para obter um hash SHA-1 de 160 bits. Alice então assina o hash com sua chave privada RSA e envia a mensagem de texto simples e o hash assinado para Bob.

SHA-1
algoritmo
H
SHA-1 de 160 bits
hash de M
D \wedge (H)
Hash assinado
RSA
algoritmo
Alice's
chave privada, D \wedge
Enviei
para
Prumo
Alice's
texto simples
mensagem
M
(arbitrário
comprimento)

Figura 8-21. Uso de SHA-1 e RSA para assinar mensagens não secretas.

Depois de receber a mensagem, Bob calcula ele próprio o hash SHA-1 e também aplica-se a chave pública de Alice para o hash assinado para obter o hash original, H . Se o dois concordam, a mensagem é considerada válida. Uma vez que não há como Trudy modificar a mensagem (texto simples) enquanto ela está em trânsito e produzir uma nova que hashes para H , Bob pode detectar facilmente quaisquer alterações que Trudy tenha feito na mensagem.

Para mensagens cuja integridade é importante, mas cujo conteúdo não é secreto, o esquema da Fig. 8-21 é amplamente utilizado. Por um custo relativamente pequeno em computação, garante que quaisquer modificações feitas na mensagem de texto simples em trânsito podem ser detectado com probabilidade muito alta.

Agora vamos ver rapidamente como o SHA-1 funciona. Ele começa preenchendo a mensagem sage adicionando 1 bit ao final, seguido por tantos 0 bits quantos forem necessários, mas pelo menos 64, para tornar o comprimento um múltiplo de 512 bits. Então um número de 64 bits con manter o comprimento da mensagem antes do preenchimento receber um OR nos 64 bits de ordem inferior. No

Fig. 8-22, a mensagem é mostrada com preenchimento à direita porque o texto em inglês e as figuras vão da esquerda para a direita (ou seja, o canto inferior direito é geralmente percebido como o final da figura). Com computadores, esta orientação corresponde ao big-endian máquinas como o SPARC e o IBM 360 e seus sucessores, mas SHA-1 al- maneiras preenchem o final da mensagem, não importa qual máquina endian é usada.

```

H2
W2
H3
Mn-1
(uma)
Início da mensagem
Bloco de 512 bits
Palavra de 32 bits
Preenchimento
(b)
(c)
H4
W79

```

Figura 8-22. (a) Uma mensagem preenchida para um múltiplo de 512 bits. (b) A saída variáveis. (c) A palavra array.

Durante o cálculo, SHA-1 mantém cinco variáveis de 32 bits, H_0 a H_4 , onde o hash se acumula. Eles são mostrados na Fig. 8-22 (b). Eles são inicializado com as constantes especificadas na norma.

Cada um dos blocos M_0 a M_{n-1} agora é processado por sua vez. Para o cur- bloco de aluguel, as 16 palavras são primeiro copiadas para o início de um auxiliar de 80 palavras matriz, W , como mostrado na Fig. 8-22 (c). Em seguida, as outras 64 palavras em W são preenchidas usando a fórmula

$$W_i = S_1 (W_{i-3} \text{ XOR } W_{i-8} \text{ XOR } W_{i-14} \text{ XOR } W_{i-16}) \quad (16 \leq i \leq 79)$$

onde $S_b (W)$ representa a rotação circular esquerda da palavra de 32 bits, W , por b bits.

Agora, cinco variáveis de zero, A a E , são inicializadas de H_0 a H_4 , respectivamente.

O cálculo real pode ser expresso em pseudo-C como

```

para (i = 0; i < 80; i++) {
    temp = S5 (A) + f
    Eu
    (B, C, D) + E + W
    Eu
    + K
    Eu
    ;
    E = D; D = C; C = S30 (B); B = A; A = temp;
}

```

onde as constantes K_i são definidas no padrão. As funções de mistura f_i são definido como

$$f_i (B, C, D) = (B \text{ E } C) \text{ OU } (\text{NÃO } B \text{ E } D) \quad (0 \leq i \leq 19)$$

$$f_i (B, C, D) = B \text{ XOR } C \text{ XOR } D \quad (20 \leq i \leq 39)$$

$$f_i (B, C, D) = (B \text{ AND } C) \text{ OR } (B \text{ AND } D) \text{ OR } (C \text{ AND } D) \quad (40 \leq i \leq 59)$$

$$f_i (B, C, D) = B \text{ XOR } C \text{ XOR } D \quad (60 \leq i \leq 79)$$

Quando todas as 80 iterações do loop forem concluídas, de A a E são adicionados a H_0 a H_4 , respectivamente.

Agora que o primeiro bloco de 512 bits foi processado, o próximo é iniciado.

A matriz W é reinicializada a partir do novo bloco, mas H permanece como estava. Quando isso

bloco é concluído, o próximo é iniciado, e assim por diante, até que toda a mensagem de 512 bits blocos foram lançados na sopa. Quando o último bloco foi concluído, o cinco palavras de 32 bits na matriz H são geradas como hash criptográfico de 160 bits. o código C completo para SHA-1 é fornecido no RFC 3174.

Novas versões de SHA-1 foram desenvolvidas para produzir hashes de 224, 256, 384 e 512 bits. Coletivamente, essas versões são chamadas de SHA-2. Não somente esses hashes são mais longos do que os hashes SHA-1, mas a função de resumo foi

alterado para combater algumas fraquezas potenciais do SHA-1. SHA-2 ainda não é amplamente usado, mas é provável que o seja no futuro.

MD5

Para completar, mencionaremos outro resumo que é popular. **MD5** (Rivest, 1992) é o quinto de uma série de resumos de mensagens projetados por Ronald Rivest. Resumidamente, a mensagem é preenchida com um comprimento de 448 bits (módulo 512). Em seguida, o comprimento original da mensagem é anexado como um inteiro de 64 bits para dar um

entrada total cujo comprimento é um múltiplo de 512 bits. Cada rodada do cálculo pega um bloco de entrada de 512 bits e mistura-o completamente com um buff de 128 bits em execução

er. Para uma boa medida, a mistura usa uma tabela construída a partir da função seno. O objetivo de usar uma função conhecida é evitar qualquer suspeita de que o designer construído em uma porta traseira inteligente pela qual só ele pode entrar. Este processo continua até que todos os blocos de entrada tenham sido consumidos. O conteúdo de 128 bits buffer do resumo da mensagem.

Depois de mais de uma década de uso e estudo sólidos, os pontos fracos no MD5 levaram à capacidade de encontrar colisões ou mensagens diferentes com o mesmo hash (Sotirov, et al., 2008). Esta é a sentença de morte para uma função digest, porque significa que o resumo não pode ser usado com segurança para representar uma mensagem. Assim, a comunidade de segurança

ity considera MD5 estar quebrado; deve ser substituído sempre que possível e nenhum novo os sistemas devem usá-lo como parte de seu projeto. No entanto, você ainda pode ver MD5 usados em sistemas existentes.

8.4.4 O Ataque de Aniversário

No mundo da criptografia, nada é o que parece ser. Alguém pode pensar que levaria na ordem de $2^{m/2}$ operações para subverter uma digitação de mensagem de m -bits est. Na verdade, operações de $2^{m/2}$ costumam funcionar usando o **ataque de aniversário**, uma abordagem

publicado por Yuval (1979) em seu artigo agora clássico " How to Swindle Rabin ".

A ideia desse ataque vem de uma técnica que os professores de matemática costumam usar em seus cursos de probabilidade. A questão é: quantos alunos você precisa em um aula antes que a probabilidade de haver duas pessoas com o mesmo aniversário exceda 1/2? A maioria dos alunos espera que a resposta seja bem acima de 100. Na verdade, a probabilidade a teoria diz que é apenas 23. Sem dar uma análise rigorosa, intuitivamente, com 23

Página 829

SEC. 8,4

ASSINATURAS DIGITAIS

805

pessoas, podemos formar $(23 \times 22) / 2 = 253$ pares diferentes, cada um dos quais tem um probabilidade de 1/365 de ser um acerto. Sob esta luz, não é tão surpreendente qualquer Mais.

De forma mais geral, se houver algum mapeamento entre entradas e saídas com n entradas (pessoas, mensagens, etc.) e k saídas possíveis (aniversários, resumos de mensagens, etc.), existem $n(n - 1)/2$ pares de entrada. Se $n(n - 1)/2 > k$, a chance de ter pelo menos uma correspondência é muito boa. Assim, aproximadamente, uma correspondência é provável para $n > \sqrt{k}$.

Este resultado significa que um resumo da mensagem de 64 bits pode provavelmente ser quebrado por geração

procurando cerca de 2^{32} mensagens e procurando duas com o mesmo resumo de mensagens.

Vejamos um exemplo prático. O Departamento de Ciência da Computação em A State University tem uma posição para um membro efetivo do corpo docente e dois candidatos datas, Tom e Dick. Tom foi contratado dois anos antes de Dick, então ele vai para revise primeiro. Se ele conseguir, Dick está sem sorte. Tom sabe que o departamento o presidente do conselho, Marilyn, tem grande consideração por seu trabalho, então pede a ela que escreva para ele

carta de recomendação ao reitor, que decidirá sobre o caso de Tom. Uma vez enviado, todas as cartas tornam-se confidenciais.

Marilyn diz a sua secretária, Ellen, para escrever uma carta ao reitor, descrevendo o que ela quer nele. Quando estiver pronto, Marilyn irá revisá-lo, calcular e assinar o Resumo de 64 bits e envie ao Reitor. Ellen pode enviar a carta mais tarde por e-mail.

Infelizmente para Tom, Ellen está romanticamente envolvida com Dick e iria gosta de fazer Tom, então ela escreve a seguinte carta com as 32 opções entre colchetes:

Caro Dean Smith,

Este [carta | mensagem] é dar o meu [honesto | franca] opinião do Prof. Tom Wilson, que é [um candidato | para cima] para estabilidade [agora | este ano]. Eu tenho [conhecido |

trabalhou com] Prof. Wilson por [sobre | quase] seis anos. Ele é um [excelente | excelente] pesquisador de grande [talento | capacidade] conhecida [em todo o mundo | internacionalmente]

por seu [brilhante | criativo] insights sobre [muitos | uma grande variedade de] [difícil | desafiar desafiadores].

Ele também é um [altamente | muito] [respeitado | admirado] [professor | educador]. Seu alunos dão suas [aulas | cursos] [rave | espetaculares] comentários. Ele é [nosso | a Departamento] [mais populares | mais amada] [professor | instrutor].

[Além disso | Além disso] o Prof. Wilson é um [talentoso | efetivo] arrecadador de fundos. Seu [concede | contratos] trouxeram um [grande | substancial] quantidade de dinheiro em [o | nosso] Departamento. [Este dinheiro tem | Esses fundos foram] [ativado | permitido] nós para [perseguir | realizar] muitos [especiais | importantes] programas, [como | para exemplo] seu programa estadual 2000. Sem esses fundos, nós [seríamos incapazes | não poder] continuar este programa, que é tão [importante | essencial] para nós dois.

Eu recomendo fortemente que você conceda a ele estabilidade.

Infelizmente para Tom, assim que Ellen terminar de escrever e digitar neste carta, ela também escreve uma segunda:

Página 830

806

SEGURANÇA DE REDE

INDIVÍDUO. 8

Caro Dean Smith,

Este [carta | mensagem] é dar o meu [honesto | franca] opinião do Prof. Tom Wilson, que é [um candidato | para cima] para estabilidade [agora | este ano]. Eu tenho [conhecido |

trabalhou com] Tom para [sobre | quase] seis anos. Ele é um [pobre | fraco] pesquisador não é muito conhecido em seu [campo | área]. Sua pesquisa [quase nunca | raramente] mostra [visão em | compreensão da] [chave | principais] problemas de [o | nosso] dia.

Além disso, ele não é um [respeitado | admirado] [professor | educador]. Seu estudantes dão suas [aulas | cursos] [pobre | más] críticas. Ele é [nosso | o Department's] menos populares [professor | instrutor], conhecido [principalmente | principalmente] dentro de [o

| nosso] Departamento para sua [tendência | propensão] para [ridículo | embaraçar] alunos [tolo | imprudente] o suficiente para fazer perguntas em suas aulas.

[Além disso | Além disso] Tom é um [pobre | marginal] arrecadador de fundos. Seu [concede | contratos] trouxeram apenas um [magro | insignificante] quantidade de dinheiro em [o | nosso] Departamento. A menos que novo [dinheiro é | fundos são] localizados rapidamente, nós

pode ser necessário cancelar alguns programas essenciais, como o programa State 2000.

Infelizmente, sob essas [condições | circunstâncias] eu não posso em boa [consciência | fé] recomendá-lo a você para [mandato | uma posição permanente].

Agora Ellen programa seu computador para computar os 2 32 resumos de mensagens de cada carta durante a noite. Provavelmente, um resumo da primeira carta corresponderá a um resumo de o segundo. Caso contrário, ela pode adicionar mais algumas opções e tentar novamente esta noite. Suponha

que ela encontra uma correspondência. Chame o 'bom' letra *A* eo 'ruim' one *B*. Ellen agora envia a carta *A* por e-mail para Marilyn para aprovação. Letra *B* que ela mantém em segredo, mostrando a ninguém. Marilyn, é claro, aprova, calcula sua mensagem de 64 bits sage digest, assina o digest e envia por e-mail o resumo assinado para Dean Smith. No-dependente, Ellen envia por e-mail a letra *B* para o reitor (não a letra *A*, como deveria). Depois de receber a carta e o resumo da mensagem assinada, o reitor executa a mensagem digerir o algoritmo da letra *B*, ver que ele concorda com o que Marilyn lhe enviou, e despede Tom. O reitor não percebe que Ellen conseguiu gerar duas cartas com o mesmo resumo de mensagem e enviou a ela um diferente do que Marilyn viu e aprovou. (Final opcional: Ellen conta a Dick o que fez. Dick é horrorizado e interrompe o caso. Ellen fica furiosa e confessa para Marilyn. Marilyn liga para o reitor. Afinal, Tom consegue a estabilidade.) Com SHA-1, o aniversário em a aderência é difícil porque, mesmo na velocidade ridícula de 1 trilhão de digere por segundo segundo, levaria mais de 32.000 anos para computar todos os 2⁸⁰ resumos de duas letras com 80 variantes cada e, mesmo assim, a correspondência não é garantida. Com uma nuvem de 1.000.000 de chips trabalhando em paralelo, 32.000 anos tornam-se 2 semanas.

8.5 GESTÃO DE CHAVES PÚBLICAS

A criptografia de chave pública torna possível para as pessoas que não compartilham um chave comum com antecedência para, no entanto, se comunicar com segurança. Também faz assinar mensagens possível sem a presença de um terceiro confiável. Finalmente,

Página 831

SEC. 8,5

GESTÃO DE CHAVES PÚBLICAS

807

resumos de mensagens assinados permitem que o destinatário verifique a integridade de recebeu mensagens com facilidade e segurança.

No entanto, há um problema que atenuamos um pouco rápido demais: se Alice e Bob não se conhecem, como eles conseguem as chaves públicas um do outro iniciar o processo de comunicação? A solução óbvia - coloque sua chave pública no seu site - não funciona, pelo seguinte motivo. Suponha que Alice deseja pesquisar a chave pública de Bob em seu site. Como ela faz isso? Ela começa digitando o URL de Bob. Seu navegador procura o endereço DNS de A página inicial de Bob e envia uma solicitação *GET*, conforme mostrado na Figura 8-23. Infeliz-Sim, Trudy intercepta a solicitação e responde com uma página inicial falsa, provavelmente um cópia da página inicial de Bob, exceto para a substituição da chave pública de Bob por Chave pública de Trudy. Quando Alice agora criptografa sua primeira mensagem com *E_T*, Trudy descriptografa, ié, criptografa novamente com a chave pública de Bob e envia para Bob, que ninguém sabe que Trudy está lendo suas mensagens recebidas. Pior ainda, Trudy poderia modificar as mensagens antes de criptografá-las novamente para Bob. Claramente, alguns é necessário um mecanismo para garantir que as chaves públicas possam ser trocadas com segurança.

4. *E_b* (mensagem)

Alice

Trudy

1. OBTER a página inicial de Bob

2. Página inicial falsa com *E_T*

3. *E_T* (mensagem)

Prumo

Figura 8-23. Uma maneira de Trudy subverter a criptografia de chave pública.

8.5.1 Certificados

Como uma primeira tentativa de distribuir chaves públicas com segurança, podemos imaginar um **Centro de distribuição de chaves KDC** disponível online 24 horas por dia para fornecer chaves sob demanda. Um dos muitos problemas com esta solução é que ela não é escalável, e o centro de distribuição de chaves rapidamente se tornaria um gargalo. Além disso, se alguma vez caísse, a segurança da Internet iria parar repentinamente. Por essas razões, as pessoas desenvolveram uma solução diferente, que faz não exigem que o centro de distribuição de chaves esteja online o tempo todo. Na verdade, não

tem que estar online. Em vez disso, o que ele faz é certificar as chaves públicas pertencentes para pessoas, empresas e outras organizações. Uma organização que certifica publicamente suas chaves é chamada de **CA (Autoridade de Certificação)**.

Por exemplo, suponha que Bob queira permitir que Alice e outras pessoas não saibam se comunicar com ele de forma segura. Ele pode ir para o CA com sua chave pública juntamente com seu passaporte ou carteira de motorista e peça para ser certificado. O CA então emite um certificado semelhante ao da Fig. 8-24 e assina seu SHA-1

Página 832

808

SEGURANÇA DE REDE INDIVÍDUO. 8

hash com a chave privada do CA. Bob então paga a taxa do CA e obtém um CD-ROM contendo o certificado e seu hash assinado.

Certifico que a chave pública

19836A8B03030CF83737E3837837FC3s87092827262643FFA827103828282A

pertence a

Robert John Smith

12345 University Avenue

Berkeley, CA 94702

Aniversário: 4 de julho de 1958

Email: bob@superduper.net.com

SHA-1 hash do certificado acima assinado com a chave privada da CA

Figura 8-24. Um possível certificado e seu hash assinado.

A tarefa fundamental de um certificado é vincular uma chave pública ao nome de um principal (pessoa física, empresa, etc.). Os certificados em si não são secretos ou protegidos. Bob pode, por exemplo, decidir colocar seu novo certificado em sua Web site, com um link na página principal dizendo: Clique aqui para obter o meu certificado de chave pública.

O clique resultante retornaria o certificado e o bloco de assinatura (o hash SHA-1 assinado do certificado).

Agora, vamos percorrer o cenário da Figura 8-23 novamente. Quando Trudy intercepta o pedido de Alice para a página inicial de Bob, o que ela pode fazer? Ela pode colocar o seu certificado e bloco de assinatura na página falsa, mas quando Alice lê o conteúdo

do certificado, ela verá imediatamente que não está falando com Bob porque

O nome de Bob não está nele. Trudy pode modificar a página inicial de Bob na hora, substituindo A chave pública de Bob com a dela. No entanto, quando Alice executa o algoritmo SHA-1 no certificado, ela receberá um hash que não concorda com o que ela obteve quando ela aplica a chave pública bem conhecida do CA ao bloco de assinatura. Desde a Trudy não tem a chave privada do CA, ela não tem como gerar um sinal bloco de estrutura que contém o hash da página da Web modificada com sua chave pública em isto. Dessa forma, Alice pode ter certeza de que possui a chave pública de Bob e não a de Trudy ou de outra pessoa. E como prometemos, este esquema não exige que o CA seja online para verificação, eliminando assim um possível gargalo.

Embora a função padrão de um certificado seja vincular uma chave pública a uma pessoa, um certificado também pode ser usado para vincular uma chave pública a um **atributo**. Para exemplo, um certificado poderia dizer: "Esta chave pública pertence a alguém com mais de 18 anos". Poderia ser usado para provar que o proprietário da chave privada não era menor de idade e, portanto,

permisão para acessar material não adequado para crianças, e assim por diante, mas sem desfechando a identidade do proprietário. Normalmente, a pessoa que possui o certificado envia-o para o site, diretor ou processo que se preocupa com a idade. Esse site, principal, ou processo geraria um número aleatório e o criptografaria com a chave pública no certificado. Se o proprietário conseguiu descriptografá-lo e enviá-lo de volta,

Página 833

SEC. 8,5

GESTÃO DE CHAVES PÚBLICAS

809

isso seria prova de que o proprietário de fato tinha o atributo declarado no certificado. Alternativamente, o número aleatório pode ser usado para gerar uma chave de sessão para a conversa seguinte.

Outro exemplo de onde um certificado pode conter um atributo é em um sistema distribuído orientado para jeto. Cada objeto normalmente possui vários métodos. o dono do objeto poderia fornecer a cada cliente um certificado dando um pouco mapa de quais métodos o cliente tem permissão para invocar e vincular o mapa de bits a uma chave pública usando um certificado assinado. Novamente, se o titular do certificado puder comprovar a posse da chave privada correspondente, terá permissão para realizar os métodos no mapa de bits. Esta abordagem tem a propriedade de que a identidade do proprietário a identidade não precisa ser conhecida, uma propriedade útil em situações em que a privacidade é importante.

8.5.2 X.509

Se todos que queriam algo assinado fossem para o CA com um diferente tipo de certificado, gerenciar todos os formatos diferentes logo se tornaria um problema lem. Para resolver este problema, um padrão para certificados foi desenvolvido e aprovado pela UIT. O padrão é denominado **X.509** e é amplamente utilizado no Internet. Ele passou por três versões desde a padronização inicial em 1988. Discutiremos a V3.

X.509 foi fortemente influenciado pelo mundo OSI, pegando emprestado alguns de seus piores recursos (por exemplo, nomenclatura e codificação). Surpreendentemente, o IETF concordou com

X.509, mesmo que em quase todas as outras áreas, de endereços de máquinas a trans-protocolos de esporte para formatos de e-mail, a IETF geralmente ignorou o OSI e tentou fazê-lo direito. A versão IETF de X.509 é descrita em RFC 5280.

Basicamente, o X.509 é uma maneira de descrever certificados. Os campos primários em um certificado estão listados na Fig. 8-25. As descrições dadas devem fornecer um ideia geral do que os campos fazem. Para obter informações adicionais, consulte o próprio padrão ou RFC 2459.

Por exemplo, se Bob trabalha no departamento de empréstimos do Money Bank, seu O endereço X.500 pode ser

/C=US /O=MoneyBank /OU=Loan /CN=Bob /
onde C é para país, O é para organização, OU é para unidade organizacional e CN é para nome comum. CAs e outras entidades são nomeadas de maneira semelhante. UMA problema substancial com nomes X.500 é que se Alice está tentando entrar em contato bob@moneybank.com e recebe um certificado com um nome X.500, pode não ser óbvio para ela que o certificado se refere ao Bob que ela deseja. Felizmente, comece com a versão 3, os nomes DNS agora são permitidos em vez de nomes X.500, portanto, problema pode eventualmente desaparecer.

Os certificados são codificados usando OSI ASN.1 (**Abstract Syntax Notation 1**), que é como uma estrutura em C, exceto com uma forma extremamente peculiar e detalhada notação. Mais informações sobre o X.509 são fornecidas por Ford e Baum (2000).

810

SEGURANÇA DE REDE
INDIVÍDUO. 8

Campo

Significado

Versão

Qual versão do X.509

Número de série

Este número mais o nome do CA identifica exclusivamente o certificado

Algoritmo de assinatura

O algoritmo usado para assinar o certificado

Emitente

Nome X.500 da CA

Período de validade

Os horários de início e término do período de validade
Nome do assunto
A entidade cuja chave está sendo certificada
Chave pública
A chave pública do assunto e o ID do algoritmo que a usa
ID do emissor
Um ID opcional que identifica exclusivamente o emissor do certificado
ID do assunto
Um ID opcional que identifica exclusivamente o assunto do certificado
Extensões
Muitas extensões foram definidas
Assinatura
A assinatura do certificado (assinado pela chave privada da CA)

Figura 8-25. Os campos básicos de um certificado X.509.

8.5.3 Infraestruturas de chave pública

Ter uma única CA para emitir todos os certificados do mundo obviamente não trabalhos. Ele entraria em colapso sob a carga e também seria um ponto central de falha. UMA solução possível pode ser ter vários CAs, todos administrados pela mesma organização e todos usando a mesma chave privada para assinar certificados. Embora isso resolvesse o problemas de carga e falha, ele apresenta um novo problema: vazamento de chave. Se lá havia dezenas de servidores espalhados pelo mundo, todos com a chave privada da CA, a chance de a chave privada ser roubada ou vazar seria aumentou muito. Uma vez que o comprometimento desta chave arruinaria o sistema eletrônico do mundo

A infraestrutura de segurança tronc, ter uma única CA central é muito arriscada.

Além disso, qual organização operaria o CA? É difícil imaginar qualquer autoridade que seja aceita mundialmente como legítima e confiável. No alguns países, as pessoas insistem que seja um governo, enquanto em outros países tenta, eles insistem que não é um governo.

Por essas razões, uma maneira diferente de certificar chaves públicas foi desenvolvida. isto recebe o nome geral de **PKI (Public Key Infrastructure)**. Neste secção, vamos resumir como funciona em geral, embora tenha havido muitos propostas, então os detalhes provavelmente irão evoluir com o tempo.

Uma PKI tem vários componentes, incluindo usuários, CAs, certificados e direc-tórias. O que a PKI faz é fornecer uma maneira de estruturar esses componentes e definir padrões para os vários documentos e protocolos. Um particularmente simples

A forma de PKI é uma hierarquia de CAs, conforme ilustrado na Figura 8-26. Neste exemplo nós mostraram três níveis, mas na prática pode haver menos ou mais. O topo-nível CA, a raiz, certifica CAs de segundo nível, que aqui chamamos de **RA s** (**Regional**

Página 835

SEC. 8,
GESTÃO DE CHAVES PÚBLICAS

811

Autoridades) porque podem cobrir alguma região geográfica, como um país ou continente. Este termo não é padrão; na verdade, nenhum termo é realmente padrão para os diferentes níveis da árvore. Estes, por sua vez, certificam os CAs reais, que emitem os certificados X.509 para organizações e indivíduos. Quando o root autoriza um novo RA, gera um certificado X.509 atestando que aprovou o RA, em clui a chave pública do novo RA nele, assina-a e a entrega ao RA. Similarmente, quando um RA aprova um novo CA, ele produz e assina um certificado declarando sua aprovação e contendo a chave pública do CA.

CA 1
CA 2
(uma)
(b)
CA 3
CA 4
CA 5
RA 2
RA 2 é aprovado.
Sua chave pública é
47383AE349...

Assinatura de Root

RA 1

Raiz

RA 2 é filho da raiz.

Sua chave pública é

47383AE4E49...

Assinatura de Root

CA 5 é aprovado.

Sua chave pública é

6384AF863B...

Assinatura de RA 2

CA 5 é aprovado.

Sua chave pública é

6384AF863B...

Assinatura de RA 2

Figura 8-26. (a) Uma PKI hierárquica. (b) Uma cadeia de certificados.

Nossa PKI funciona assim. Suponha que Alice precise da chave pública de Bob para se comunicar com ele, então ela procura e encontra um certificado que o contenha, assinado por CA 5. Mas Alice nunca ouviu falar de CA 5. Pelo que ela sabe, CA 5 pode ser a filha de 10 anos de Bob. Ela poderia ir ao CA 5 e dizer: "Prove seu legitimidade." CA 5 responderá com o certificado que obteve de RA 2, que contém a chave pública do CA 5. Agora armado com a chave pública do CA 5, ela pode verificar se o certificado de Bob foi realmente assinado pela CA 5 e, portanto, é legal. A menos que RA 2 seja o filho de 12 anos de Bob. Então, o próximo passo é ela pedir ao RA 2 para provar que é legítimo. A resposta à sua pergunta é um certificado assinado pelo root e contendo a chave pública de RA 2. Agora Alice tem certeza de que possui a chave pública de Bob.

Mas como Alice encontra a chave pública da raiz? Magia. É assumido que todo mundo conhece a chave pública do root. Por exemplo, seu navegador pode ter sido enviado com a chave pública do root embutida.

Bob é um cara amigável e não quer causar muito trabalho a Alice.

Ele sabe que ela terá que verificar CA 5 e RA 2, para salvá-la

algum problema, ele coleta os dois certificados necessários e dá a ela os dois certificados juntos com o dele. Agora ela pode usar seu próprio conhecimento do público do root chave para verificar o certificado de nível superior e a chave pública nele contida para verificar o segundo. Alice não precisa entrar em contato com ninguém para fazer a verificação.

Página 836

812

SEGURANÇA DE REDE

INDIVÍDUO. 8

Como os certificados são todos assinados, ela pode detectar facilmente qualquer tentativa de tamponar com seus conteúdos. Uma cadeia de certificados voltando à raiz como esta é às vezes chamada de **cadeia de confiança** ou **caminho de certificação**. A técnica é amplamente usada na prática.

Claro, ainda temos o problema de quem vai executar o root. A solução não é ter uma única raiz, mas ter muitas raízes, cada uma com suas RAs e CAs. Na verdade, os navegadores modernos vêm pré-carregados com as chaves públicas para mais de 100 raízes, às vezes chamadas de **âncoras de confiança**. Desta forma, tendo um pecado Uma autoridade confiável mundial pode ser evitada.

Mas agora há a questão de como o fornecedor do navegador decide qual suposto as âncoras de confiança são confiáveis e desprezíveis. Tudo depende do usuário confiar no fornecedor do navegador para fazer escolhas sábias e não simplesmente aprovar toda confiança

âncoras dispostas a pagar sua taxa de inclusão. A maioria dos navegadores permite que os usuários inspecionem suas chaves raiz (geralmente na forma de certificados assinados pela raiz) e exclua qualquer que parecem sombrios.

Diretórios

Outro problema para qualquer PKI é onde os certificados (e suas cadeias de volta para algumas âncoras de confiança conhecidas) são armazenados. Uma possibilidade é fazer com que cada usuário armazene seu ou seus próprios certificados. Embora seja seguro fazer isso (ou seja, não há como os usuários adulterar certificados assinados sem detecção), também é inconveniente. Um alternativa proposta é usar o DNS como um diretório de certificados. Antes

contatando Bob, Alice provavelmente terá que procurar seu endereço IP usando DNS, então por que o DNS não retornou toda a cadeia de certificados de Bob junto com seu endereço IP? Algumas pessoas pensam que este é o caminho a percorrer, mas outros preferem diretores de reitoria cujo único trabalho é gerenciar certificados X.509. Esses diretórios poderia fornecer serviços de pesquisa usando propriedades dos nomes X.500. Para exemplo, em teoria, tal serviço de diretório poderia responder a uma pergunta como: "Dê-me uma lista de todas as pessoas chamadas Alice que trabalham em departamentos de vendas em qualquer lugar do EUA ou Canadá."

Revogação

O mundo real também está cheio de certificados, como passaportes e motoristas licenças. Às vezes, esses certificados podem ser revogados, por exemplo, drivers' licenças podem ser revogadas para dirigir embriagado e outras infrações de condução. O mesmo problema ocorre no mundo digital: o concedente de um certificado pode decidir revogá-lo porque a pessoa ou organização que o detém abusou dele de alguma forma. Também pode ser revogado se a chave privada do assunto foi exposta ou, pior ainda, a chave privada da CA foi comprometida. Assim, uma PKI precisa lidar com a questão de revogação. A possibilidade de revogação complica as coisas.

Página 837

SEC. 8,5

GESTÃO DE CHAVES PÚBLICAS

813

Um primeiro passo nessa direção é fazer com que cada CA emita periodicamente uma **CRL** (**Lista de revogação de certificado**) fornecendo os números de série de todos os certificados que revogou. Uma vez que os certificados contêm tempos de expiração, a CRL precisa apenas conter os números de série dos certificados que ainda não expiram. Uma vez que seu tempo de expiração passou, um certificado é automaticamente inválido, então nenhuma distinção é necessária entre aqueles que acabaram de expirar e aqueles que foram realmente revogados. Em ambos casos, eles não podem mais ser usados.

Infelizmente, a introdução de CRLs significa que um usuário que está prestes a usar um certificado deve agora adquirir a CRL para ver se o certificado foi revogado. Se isso foi, não deve ser usado. No entanto, mesmo que o certificado não esteja na lista, ela pode ter sido revogada logo após a publicação da lista. Assim, a única maneira de realmente certifique-se de perguntar ao CA. E no próximo uso do mesmo certificado, o CA deve ser perguntado novamente, uma vez que o certificado pode ter sido revogado alguns segundos atrás.

Outra complicação é que um certificado revogado poderia ser rein-declarado, por exemplo, se foi revogado por falta de pagamento de alguma taxa que desde então foi pago. Ter que lidar com a revogação (e possivelmente a reintegração) elimina uma das melhores propriedades dos certificados, ou seja, que eles podem ser usados sem ter que entrar em contato com um CA.

Onde as CRLs devem ser armazenadas? Um bom lugar seria o mesmo lugar que os próprios certificados são armazenados. Uma estratégia é o CA promoverativamente CRLs periodicamente e os diretórios os processam simplesmente removendo o certificados revogados. Se os diretórios não forem usados para armazenar certificados, as CRLs pode ser armazenado em cache em vários locais da rede. Uma vez que uma CRL é ela própria um documento assinado, se for adulterado, essa adulteração pode ser facilmente detectada.

Se os certificados têm vida útil longa, as CRLs também serão longas. Por exemplo, se os cartões de crédito são válidos por 5 anos, o número de revogações pendentes será muito mais do que se novos cartões fossem emitidos a cada 3 meses. Uma maneira padrão de lidar com CRLs longas é emitir uma lista mestre com pouca frequência, mas emitir atualizações para ela mais frequentemente. Isso reduz a largura de banda necessária para distribuir as CRLs.

8.6 SEGURANÇA DE COMUNICAÇÃO

Agora terminamos nosso estudo das ferramentas do ofício. A maioria dos importantes tópicos técnicos e protocolos foram cobertos. O resto do capítulo é sobre

como essas técnicas são aplicadas na prática para fornecer segurança de rede, além de algumas reflexões sobre os aspectos sociais da segurança no final do capítulo. Nas quatro seções a seguir, veremos a segurança da comunicação, ou seja, como obter os bits secretamente e sem modificação da origem ao destino e como manter pedaços indesejados fora da porta. Estes não são de forma alguma os únicos problemas de segurança na rede, mas certamente estão entre os mais importantes alguns, tornando este um bom lugar para começar nosso estudo.

Página 838

814

SEGURANÇA DE REDE
INDIVÍDUO. 8

8.6.1 IPsec

A IETF sabia há anos que faltava segurança na Internet. Adicionando não foi fácil porque eclodiu uma guerra sobre onde colocá-lo. A maioria dos especialistas em segurança acreditam que para serem realmente seguros, as verificações de criptografia e integridade devem ser feitas final (ou seja, na camada de aplicativo). Ou seja, o processo de origem criptografa e / ou integridade protege os dados e os envia para o processo de destino onde eles estão descriptografado e / ou verificado. Qualquer adulteração feita entre esses dois processos, incluindo dentro de qualquer sistema operacional, pode então ser detectado. O problema com esta abordagem é que requer a mudança de todos os aplicativos para torná-los seguros ty ciente. Nesta visão, a próxima melhor abordagem é colocar criptografia no trans- camada de esporte ou em uma nova camada entre a camada de aplicação e a camada de transporte, tornando-o ainda de ponta a ponta, mas não exigindo que os aplicativos sejam alterados. A visão oposta é que os usuários não entendem de segurança e não serão capaz de usá-lo corretamente e ninguém quer modificar os programas existentes em qualquer forma, então a camada de rede deve autenticar e / ou criptografar pacotes sem o usuários envolvidos. Depois de anos de batalhas campais, essa visão ganhou sup- porta que um padrão de segurança da camada de rede foi definido. Em parte, o argumento era que ter a criptografia da camada de rede não impede que usuários clientes da segurança fazê-lo da maneira certa e até certo ponto ajuda usuários desatentos à segurança. O resultado desta guerra foi um design chamado **IPsec (segurança IP)**, que é de- scrito nas RFCs 2401, 2402 e 2406, entre outros. Nem todos os usuários querem entrar criptografia (porque é computacionalmente cara). Em vez de torná-lo opcional, foi decidido exigir criptografia o tempo todo, mas permitir o uso de um algoritmo nulo ritmo. O algoritmo nulo é descrito e elogiado por sua simplicidade, facilidade de implementação e grande velocidade no RFC 2410. O design IPsec completo é uma estrutura para vários serviços, algoritmos, e granularidades. A razão para vários serviços é que nem todo mundo quer pagar o preço por ter todos os serviços o tempo todo, para que os serviços estejam disponíveis a o menu. Os principais serviços são sigilo, integridade de dados e proteção contra ataques de repetição (onde o intruso repete uma conversa). Todos estes são baseados na criptografia de chave simétrica porque o alto desempenho é crucial. A razão para ter vários algoritmos é que um algoritmo que agora é pensado para ser seguro pode ser quebrado no futuro. Ao tornar o algoritmo IPsec em dependente, a estrutura pode sobreviver mesmo se algum algoritmo particular for posterior partido. A razão para ter várias granularidades é tornar possível proteger um única conexão TCP, todo o tráfego entre um par de hosts ou todo o tráfego entre um par de roteadores seguros, entre outras possibilidades. Um aspecto ligeiramente surpreendente do IPsec é que, embora esteja na camada IP, é orientado para conexão. Na verdade, isso não é tão surpreendente porque ter qualquer segurança, uma chave deve ser estabelecida e usada por algum período de tempo - em essência, um tipo de conexão com um nome diferente. Além disso, as conexões amortizam a configuração

SEC. 8,6
SEGURANÇA DE COMUNICAÇÃO

815

custos sobre muitos pacotes. Uma "conexão" no contexto de IPsec é chamada de **SA** (**Associação de Segurança**). Um SA é uma conexão simplex entre dois pontos de extremidade e tem um identificador de segurança associado a ele. Se o tráfego seguro for necessário em ambos instruções, duas associações de segurança são necessárias. Identificadores de segurança são transportados

em pacotes que viajam nessas conexões seguras e são usados para procurar chaves e outras informações relevantes quando um pacote seguro chega.

Tecnicamente, o IPsec tem duas partes principais. A primeira parte descreve dois novos cabeçalhos que podem ser adicionados aos pacotes para transportar o identificador de segurança, integridade con-

dados trol e outras informações. A outra parte, **ISAKMP (Internet Security Associação e Protocolo de Gerenciamento de Chaves)**, trata do estabelecimento de chaves.

ISAKMP é um framework. O protocolo principal para a realização do trabalho é o **IKE** (**Internet Key Exchange**). A versão 2 do IKE, conforme descrito no RFC 4306, deve ser usado, já que a versão anterior era profundamente falha, conforme apontado por Perlman e Kaufman (2000).

O IPsec pode ser usado em qualquer um dos dois modos. No **modo de transporte**, o IPsec o cabeçalho é inserido logo após o cabeçalho IP. O campo *Protocolo* no cabeçalho IP é alterado para indicar que um cabeçalho IPsec segue o cabeçalho IP normal (antes do Cabeçalho TCP). O cabeçalho IPsec contém informações de segurança, principalmente o SA identificador, um novo número de sequência e possivelmente uma verificação de integridade da carga útil.

No **modo túnel**, todo o pacote IP, cabeçalho e tudo, é encapsulado no corpo de um novo pacote IP com um cabeçalho IP completamente novo. O modo túnel é útil quando o túnel termina em um local diferente do destino final. Em alguns casos, o fim do túnel é uma máquina de gateway de segurança, por exemplo, um incêndio da empresa parede. Este é normalmente o caso de uma VPN (Rede Privada Virtual). Nisso modo, o gateway de segurança encapsula e desencapsula os pacotes conforme eles passam através dele. Ao terminar o túnel nesta máquina segura, as máquinas no a LAN da empresa não precisa estar ciente do IPsec. Apenas o gateway de segurança tem para saber sobre isso.

O modo de túnel também é útil quando um pacote de conexões TCP é agregado e tratado como um fluxo criptografado porque evita que um intruso veja quem está enviando quantos pacotes para quem. Às vezes, apenas sabendo quanto o tráfego está indo para onde há informações valiosas. Por exemplo, se durante um período militar crise, a quantidade de tráfego fluindo entre o Pentágono e a Casa Branca deviam cair drasticamente, mas a quantidade de tráfego entre o Pentágono e algumas instalação militar nas profundezas das Montanhas Rochosas do Colorado deveriam aumentar em a mesma quantidade, um intruso pode ser capaz de deduzir algumas informações úteis a partir desses dados. Estudar os padrões de fluxo de pacotes, mesmo se eles estiverem criptografados,

é chamado de **análise de tráfego**. O modo de túnel fornece uma maneira de frustrá-lo até certo ponto.

A desvantagem do modo de túnel é que adiciona um cabeçalho de IP extra, aumentando assim tamanho do pacote substancialmente. Em contraste, o modo de transporte não afeta o tamanho do pacote tanto quanto.

O primeiro novo cabeçalho é **AH** (**Authentication Header**). Fornece integridade verificação e segurança anti-reprodução, mas não sigilo (ou seja, sem criptografia de dados). o

SEGURANÇA DE REDE INDIVÍDUO. 8

o uso de AH no modo de transporte é ilustrado na Fig. 8-27. No IPv4, é interposto entre o cabeçalho IP (incluindo quaisquer opções) e o cabeçalho TCP. No IPv6, é apenas mais um cabeçalho de extensão e é tratado como tal. Na verdade, o formato é próximo de aquele de um cabeçalho de extensão IPv6 padrão. A carga útil pode ter que ser preenchida a algum comprimento específico para o algoritmo de autenticação, conforme mostrado.

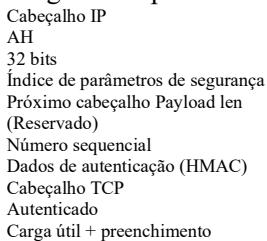


Figura 8-27. O cabeçalho de autenticação IPsec no modo de transporte para IPv4.

Vamos agora examinar o cabeçalho AH. O *próximo* campo de *cabeçalho* é usado para armazenar o valor que o campo do *protocolo* IP tinha antes de ser substituído por 51 para indicar que segue-se um cabeçalho AH. Na maioria dos casos, o código para TCP (6) irá aqui. O *comprimento da carga útil* é o número de palavras de 32 bits no cabeçalho AH menos 2. O *índice de parâmetros de segurança* é o identificador de conexão. É inserido por o remetente para indicar um determinado registro no banco de dados do receptor. Este registro contém a chave compartilhada usada nesta conexão e outras informações sobre o conexão. Se este protocolo tivesse sido inventado pela ITU em vez da IETF, este campo teria sido chamado de *número de circuito virtual*.

O campo *Sequence number* é usado para numerar todos os pacotes enviados em um SA. Cada pacote recebe um número único, até mesmo retransmissões. Em outras palavras, o retransmissão de um pacote obtém um número diferente aqui do que o original (mesmo embora seu número de sequência TCP seja o mesmo). O objetivo deste campo é detectar ataques de repetição. Esses números de sequência não podem ser agrupados. Se todos 2³² estiverem esgotados, um novo SA deve ser estabelecido para continuar a comunicação.

Finalmente, chegamos aos *dados de autenticação*, que é um campo de comprimento variável que contém a assinatura digital da carga útil. Quando o SA é estabelecido, os dois os lados negociam qual algoritmo de assinatura usarão. Normalmente, pub-

A criptografia de chave lic não é usada aqui porque os pacotes devem ser processados ao extremo rapidamente e todos os algoritmos de chave pública conhecidos são muito lentos. Como o IPsec é baseado

na criptografia de chave simétrica e o remetente e o destinatário negociam um compartilhamento antes de configurar um SA, a chave compartilhada é usada no cálculo da assinatura.

Uma maneira simples é calcular o hash sobre o pacote mais a chave compartilhada. o a chave compartilhada não é transmitida, é claro. Um esquema como este é chamado de **HMAC**

Página 841

SEC. 8,6 SEGURANÇA DE COMUNICAÇÃO

817

(**Código de autenticação de mensagem com hash**). É muito mais rápido calcular do que primeiro executando SHA-1 e, em seguida, executando RSA no resultado.

O cabeçalho AH não permite a criptografia dos dados, por isso é muito útil quando a verificação de integridade é necessária, mas o sigilo não é necessário. Uma característica notável

A característica do AH é que a verificação de integridade cobre alguns dos campos do cabeçalho IP, ou seja, aqueles que não mudam conforme o pacote se move de um roteador para outro. o *Time to live* muda em cada salto, por exemplo, por isso não pode ser incluído em a verificação de integridade. No entanto, o endereço IP de origem está incluído na verificação, tornando impossível para um intruso falsificar a origem de um pacote.

O cabeçalho IPsec alternativo é **ESP** (**Encapsulating Security Payload**). Está o uso para o modo de transporte e modo de túnel é mostrado na Figura 8-28.

```

ESP
cabeçalho
Novo IP
cabeçalho
IP antigo
cabeçalho
TCP
cabeçalho
Autenticado
Carga útil + preenchimento
(b)
Autenticação (HMAC)
ESP
cabeçalho
IP
cabeçalho
TCP
cabeçalho
Carga útil + preenchimento
(uma)
Autenticação (HMAC)
Autenticado
Criptografado
Criptografado

```

Figura 8-28. (a) ESP no modo de transporte. (b) ESP em modo de túnel.

O cabeçalho ESP consiste em duas palavras de 32 bits. Eles são o *parâmetro de segurança* campos de *índice* e *número de sequência* que vimos no AH. Uma terceira palavra que geralmente segue-os (mas tecnicamente não faz parte do cabeçalho) é a *inicialização vetor* usado para a criptografia de dados, a menos que seja usada criptografia nula, caso em que é omitido.

ESP também fornece verificações de integridade HMAC, assim como AH, mas em vez de sendo incluídos no cabeçalho, eles vêm após a carga útil, conforme mostrado na Fig. 8-28. Colocar o HMAC no final tem uma vantagem em uma implementação de hardware: o O HMAC pode ser calculado conforme os bits estão saindo pela interface de rede e anexado ao final. É por isso que Ethernet e outras LANs têm seus CRCs em um trailer, em vez de em um cabeçalho. Com AH, o pacote deve ser armazenado em buffer e o assinatura computada antes que o pacote possa ser enviado, potencialmente reduzindo o número de pacotes / s que podem ser enviados.

Dado que ESP pode fazer tudo o que AH pode fazer e muito mais e é mais eficiente para inicializar, surge a pergunta: por que se preocupar em ter AH? A resposta é principalmente histórico. Originalmente, o AH tratava apenas da integridade e o ESP tratava apenas do sigilo. Mais tarde, integridade foi adicionada ao ESP, mas as pessoas que projetaram AH não quiseram deixá-lo morrer depois de todo aquele trabalho. Seu único argumento real é que AH verifica parte do o cabeçalho IP, o que o ESP não faz, mas fora isso é realmente um argumento fraco. Outro argumento fraco é que um produto que suporta AH, mas não ESP pode

Página 842

818

SEGURANÇA DE REDE INDIVÍDUO. 8

terá menos problemas para obter uma licença de exportação porque não pode fazer criptografia. AH é provavelmente serão eliminados no futuro.

8.6.2 Firewalls

A capacidade de conectar qualquer computador, em qualquer lugar, a qualquer outro computador, qualquer

onde, é uma bênção mista. Para os indivíduos em casa, vagando pelo Internet é muito divertido. Para gerentes de segurança corporativa, é um pesadelo. Mais com empresas têm grandes quantidades de informações confidenciais online - segredos comerciais, produtos

planos de desenvolvimento de produtos, estratégias de marketing, análises financeiras, etc. Divulgação de

essas informações para um concorrente podem ter consequências terríveis.

Além do perigo de vazamento de informações, também existe o perigo de vazamento de informações. Em particular, vírus, worms e outras pragas digitais podem

violar a segurança, destruir dados valiosos e desperdiçar grandes quantidades de administradores hora de tentar limpar a bagunça que eles deixam. Muitas vezes, eles são importados por descuidados funcionários que querem jogar algum novo jogo bacana.

Consequentemente, são necessários mecanismos para manter os bits "bons" e os bits "ruins" Fora. Um método é usar o IPsec. Esta abordagem protege os dados em trânsito entre sites seguros. No entanto, o IPsec não faz nada para impedir que pragas e invasores digitais entrar na LAN da empresa. Para ver como cumprir esse objetivo, precisamos olhe para firewalls.

Os firewalls são apenas uma adaptação moderna daquela velha reserva de segurança medieval: cavar um fosso profundo ao redor de seu castelo. Este projeto forçou todos a entrar ou deixando o castelo para passar por uma única ponte levadiça, onde poderiam ser inspecionados pela polícia de I / O. Com redes, o mesmo truque é possível: uma empresa pode ter muitas LANs conectadas de forma arbitrária, mas todo o tráfego de ou para a empresa é forçada através de uma ponte levadiça eletrônica (firewall), conforme mostrado na Figura 8-29. Não outra rota existe.

Rede interna
Zona desmilitarizada
Externo
Internet
O email
servidor
Rede
servidor
Segurança
perímetro
Firewall

Figura 8-29. Um firewall protegendo uma rede interna.

Página 843

SEC. 8,6 SEGURANÇA DE COMUNICAÇÃO

819

O firewall atua como um **filtro de pacotes**. Ele inspeciona cada entrada e pacote de saída. Pacotes que atendem a alguns critérios descritos nas regras formuladas por o administrador da rede são encaminhados normalmente. Aqueles que falham no teste são caiu sem cerimônia.

O critério de filtragem é normalmente dado como regras ou tabelas que listam fontes e destinos que são aceitáveis, fontes e destinos que estão bloqueados, e des regras de falha sobre o que fazer com os pacotes vindos ou indo para outras máquinas.

No caso comum de uma configuração TCP / IP, uma origem ou destino pode consistir em um endereço IP e uma porta. As portas indicam qual serviço é desejado. Por exemplo, A porta 25 TCP é para e-mail e a porta 80 TCP é para HTTP. Algumas portas podem ser simplesmente

bloqueado. Por exemplo, uma empresa pode bloquear pacotes de entrada para todos os anúncios IP vestidos combinados com porta TCP 79. Já foi popular para o serviço Finger para procure os endereços de e-mail das pessoas, mas é pouco usado hoje.

Outras portas não são bloqueadas tão facilmente. A dificuldade é que a rede administra os tratadores querem segurança, mas não podem cortar a comunicação com o mundo exterior.

Esse arranjo seria muito mais simples e melhor para a segurança, mas haveria não haverá fim para as reclamações dos usuários sobre isso. É aqui que arranjos como o

A DMZ (Zona Desmilitarizada) mostrada na Fig. 8-29 é útil. O DMZ é o parte da rede da empresa que fica fora do perímetro de segurança. Qualquer coisa Vai aqui. Ao colocar uma máquina como um servidor da Web na DMZ, os computadores a Internet pode contatá-lo para navegar no site da empresa. Agora o firewall pode ser configurado para bloquear o tráfego TCP de entrada para a porta 80 para que os computadores no A Internet não pode usar essa porta para atacar computadores na rede interna. Permitir o servidor web a ser gerenciado, o firewall pode ter uma regra para permitir conexões entre máquinas internas e o servidor web.

Os firewalls se tornaram muito mais sofisticados ao longo do tempo em uma corrida armamentista com atacantes. Originalmente, os firewalls aplicavam um conjunto de regras independente para cada pacote, mas provou ser difícil escrever regras que permitiam funcionalidades úteis, mas

bloqueou todo o tráfego indesejado. **Firewalls com estado** mapeiam pacotes para conexões e usam campos de cabeçalho TCP / IP para controlar as conexões. Isso permite regras que, por exemplo, permitir que um servidor Web externo envie pacotes para um host interno, mas apenas se o host interno primeiro estabelecer uma conexão com o servidor da Web externo ver. Essa regra não é possível com projetos sem estado que devem ser aprovados ou abandonados todos os pacotes do servidor Web externo.

Outro nível de sofisticação do processamento stateful é para o firewall para implementar **gateways no nível do aplicativo**. Este processamento envolve o firewall olhando dentro dos pacotes, além até mesmo do cabeçalho TCP, para ver o que o aplicativo está fazendo. Com esse recurso, é possível distinguir o tráfego HTTP usado para Navegação na Web a partir do tráfego HTTP usado para compartilhamento de arquivos ponto a ponto. Administradores podem escrever regras para poupar a empresa do compartilhamento de arquivos ponto a ponto, mas permitir A navegação na Web é vital para os negócios. Para todos esses métodos, o tráfego de saída pode ser inspecionado, bem como o tráfego de entrada, por exemplo, para evitar documentos sejam enviados por e-mail para fora da empresa.

Página 844

820

SEGURANÇA DE REDE INDIVÍDUO. 8

Como a discussão acima deve deixar claro, os firewalls violam o padrão layering de protocolos. Eles são dispositivos da camada de rede, mas espiam o transporte e camadas de aplicativos para fazer sua filtragem. Isso os torna frágeis. Para exemplo, firewalls tendem a confiar em convenções de numeração de porta padrão para determinar mina que tipo de tráfego é transportado em um pacote. Portas padrão são freqüentemente usadas, mas

nem por todos os computadores, nem por todos os aplicativos. Alguns aplicativos ponto a ponto cátions selecionam portas dinamicamente para evitar serem facilmente detectados (e bloqueados). En-

criptografia com IPSEC ou outros esquemas oculta as informações da camada superior do firewall. Finalmente, um firewall não pode se comunicar prontamente com os computadores que se comunicam

através dele para dizer a eles quais políticas estão sendo aplicadas e por que sua conexão é sendo descartado. Deve simplesmente fingir ser um fio quebrado. Por todas essas razões, puristas de rede consideram os firewalls uma mancha na arquitetura do Internet. No entanto, a Internet pode ser um lugar perigoso se você for um computador.

Os firewalls ajudam com esse problema, então é provável que continuem.

Mesmo que o firewall esteja perfeitamente configurado, muitos problemas de segurança ainda existir. Por exemplo, se um firewall estiver configurado para permitir pacotes de apenas redes específicas (por exemplo, as outras fábricas da empresa), um intruso fora do firewall pode inserir endereços de origem falsos para ignorar essa verificação. Se um insider deseja enviar documentos secretos, ele pode criptografá-los ou até mesmo fotografá-los e enviar fotos como arquivos JPEG, o que ignora quaisquer filtros de e-mail. E não temos nem discutiu o fato de que, embora três quartos de todos os ataques venham de fora o firewall, os ataques que vêm de dentro do firewall, por exemplo, de disfuncionários grunhidos são normalmente os mais prejudiciais (Verizon, 2009).

Um problema diferente com firewalls é que eles fornecem um único perímetro de defesa. Se essa defesa for violada, todas as apostas serão canceladas. Por este motivo, os firewalls são

frequentemente usado em uma defesa em camadas. Por exemplo, um firewall pode proteger a entrada para

a rede interna e cada computador também pode executar seu próprio firewall. Leitores que pensam que um ponto de verificação de segurança é suficiente, claramente não fizeram um intervo nacional em uma companhia aérea regular recentemente.

Além disso, existe toda uma outra classe de ataques que os firewalls não conseguem lidar

com. A ideia básica de um firewall é impedir que intrusos entrem e se sigam dados de sair. Infelizmente, existem pessoas que não têm nada melhor para fazer do que tentar derrubar certos sites. Eles fazem isso enviando pacotes legítimos no alvo em grande número até que ele desmorone sob a carga. Por exemplo, para incapacitar um site, um intruso pode enviar um pacote TCP SYN para estabelecer uma connection. O site irá então alocar um slot de mesa para a conexão e enviar um SYN + Pacote ACK em resposta. Se o intruso não responder, o slot da mesa será empurrado por alguns segundos até o tempo limite. Se o intruso enviar milhares de solicitações de conexão, todos os slots da mesa serão preenchidos e nenhuma conexão legítima será capaz de passar. Ataques em que o objetivo do invasor é desligar o alvejar em vez de roubar dados são chamados de **ataques DoS (Denial of Service)**. Geralmente, os pacotes de solicitação têm endereços de origem falsos, então o intruso não pode ser rastreado facilmente. Os ataques DoS contra os principais sites da Web são comuns na Internet.

Página 845

SEC. 8,6
SEGURANÇA DE COMUNICAÇÃO

821

Uma variante ainda pior é aquela em que o intruso já invadiu centenas de computadores em outras partes do mundo, e então comanda todos eles para ataque o mesmo alvo ao mesmo tempo. Essa abordagem não apenas aumenta o poder de fogo do intruso, mas também reduz suas chances de detecção, uma vez que os pacotes vêm de um grande número de máquinas pertencentes a usuários desavisados. Esse tipo de ataque é chamado de ataque **DDoS (Negação de Serviço Distribuída)**. Este em-aderência é difícil de se defender. Mesmo se a máquina atacada puder rapidamente reconhecer uma solicitação falsa, leva algum tempo para processar e descartar a busca, e se chegarem solicitações suficientes por segundo, a CPU gastará todo o seu tempo lidar com eles.

8.6.3 Redes Privadas Virtuais

Muitas empresas têm escritórios e fábricas espalhadas por muitas cidades, algumas vezes em vários países. Antigamente, antes das redes públicas de dados, era comum que essas empresas alugassem linhas da companhia telefônica antes de entre alguns ou todos os pares de locais. Algumas empresas ainda fazem isso. Uma rede construída a partir de computadores da empresa e arrendadas linhas de telefone é chamado de **privada rede**.

As redes privadas funcionam bem e são muito seguras. Se as únicas linhas disponíveis forem as linhas alugadas, nenhum tráfego pode vaziar para fora dos locais da empresa e os intrusos precisam fisicamente grampear as linhas para invadir, o que não é fácil de fazer. O problema com redes privadas é que alugar uma linha T1 dedicada entre dois pontos custa milhares de dólares por mês e as linhas T3 são muitas vezes mais caras. Quando redes públicas de dados e mais tarde a Internet apareceu, muitas empresas queriam mover seu tráfego de dados (e possivelmente de voz) para a rede pública, mas sem dar aumentar a segurança da rede privada.

Essa demanda logo levou à invenção de **VPNs (Redes Privadas Virtuais)**, que são redes sobrepostas em cima de redes públicas, mas com a maioria das laços de redes privadas. Eles são chamados de "virtuais" porque são apenas um ilusão, assim como os circuitos virtuais não são circuitos reais e a memória virtual não é real memória.

Uma abordagem popular é construir VPNs diretamente sobre a Internet. Um comum design é equipar cada escritório com um firewall e criar túneis através do Inter rede entre todos os pares de escritórios, conforme ilustrado na Fig. 8-30 (a). Uma outra vantagem de usar a Internet para conectividade é que os túneis podem ser configurados sob demanda incluir, por exemplo, o computador de um funcionário que está em casa ou viajando contanto que a pessoa tenha uma conexão com a Internet. Essa flexibilidade é muito maior então é fornecido com linhas alugadas, ainda da perspectiva dos computadores em

a VPN, a topologia se parece com o caso da rede privada, conforme mostrado em Fig. 8-30 (b). Quando o sistema é ativado, cada par de firewalls deve negociar estabelecer os parâmetros de seu SA, incluindo os serviços, modos, algoritmos e chaves. Se o IPsec for usado para o túnel, é possível agregar todo o tráfego entre qualquer

Página 846

822

SEGURANÇA DE REDE
INDIVÍDUO. 8

Casa
Internet
Paris
escritório
Londres
escritório
Viagem
Casa
Viagem
Londres
Paris
(uma)
(b)

Figura 8-30. (a) Uma rede privada virtual. (b) Topologia vista de dentro.

dois pares de escritórios em um único SA criptografado e autenticado, fornecendo assim controle de integridade, sigilo e até uma imunidade considerável à análise de tráfego. Muitos firewalls têm recursos de VPN integrados. Alguns roteadores comuns podem fazer isso como bem, mas como os firewalls são principalmente no ramo de segurança, é natural ter os túneis começam e terminam nos firewalls, proporcionando uma separação clara entre a empresa e a Internet. Assim, firewalls, VPNs e IPsec com ESP no tun-

O modo nel é uma combinação natural e amplamente utilizado na prática.

Depois que as SAs forem estabelecidas, o tráfego pode começar a fluir. Para um roteador na Internet, um pacote viajando ao longo de um túnel VPN é apenas um pacote comum pacote. A única coisa incomum sobre isso é a presença do cabeçalho IPsec após o cabeçalho IP, mas como esses cabeçalhos extras não têm efeito no processo de encaminhamento Portanto, os roteadores não se importam com esse cabeçalho extra.

Outra abordagem que está ganhando popularidade é fazer com que o ISP configure a VPN.

Usando MPLS (conforme discutido no Capítulo 5), os caminhos para o tráfego VPN podem ser configurados a-

cruzar a rede ISP entre os escritórios da empresa. Esses caminhos mantêm a VPN tráfego separado de outro tráfego da Internet e pode ser garantido uma certa quantidade de largura de banda ou outra qualidade de serviço.

A principal vantagem de uma VPN é que ela é completamente transparente para todos os usuários de software

porcelana. Os firewalls configuram e gerenciam os SAs. A única pessoa que é justa ciente desta configuração é o administrador do sistema que deve configurar e gerenciar os gateways de segurança ou o administrador do ISP que deve configurar o MPLS caminhos. Para todos os outros, é como ter uma rede privada de linha alugada novamente. Para mais sobre VPNs, consulte Lewis (2006).

8.6.4 Segurança sem fio

É surpreendentemente fácil projetar um sistema usando VPNs e firewalls que são log totalmente seguro, mas que, na prática, vaza como uma peneira. Esta situação pode ocorrem se algumas das máquinas são sem fio e usam comunicação de rádio, que passa direto pelo firewall em ambas as direções. O alcance das redes 802.11 é

Página 847

SEC. 8,6
SEGURANÇA DE COMUNICAÇÃO

823

muitas vezes algumas centenas de metros, então qualquer pessoa que queira espionar uma empresa pode sim-

dirija até o estacionamento dos funcionários pela manhã, deixe um dispositivo habilitado para 802.11

notebook no carro para gravar tudo o que ouve, e decolar para o dia. No final da tarde, o disco rígido estará cheio de guloseimas valiosas. Teoricamente, esse vazamento não deveria acontecer. Teoricamente, as pessoas não são supostas para roubar bancos, também.

Muito do problema de segurança pode ser atribuído aos fabricantes de sistemas sem fio estações base (pontos de acesso) tentando tornar seus produtos amigáveis ao usuário. Geralmente, se o usuário tirar o dispositivo da caixa e conectá-lo à energia elétrica socket, ele começa a operar imediatamente - quase sempre sem nenhuma segurança, revelando segredos para todos dentro do alcance do rádio. Se estiver conectado a um Ethernet, todo o tráfego Ethernet aparece repentinamente no estacionamento também. Sem fio é o sonho de qualquer espião: dados livres sem ter que fazer nenhum trabalho. Lá-portanto, nem é preciso dizer que a segurança é ainda mais importante para sistemas sem fio do que para os com fio. Nesta seção, veremos algumas maneiras pelas quais as redes sem fio lidam com a segurança. Algumas informações adicionais são fornecidas por Nichols e Lekkas (2002).

Segurança 802.11

Parte do padrão 802.11, originalmente chamado de **802.11i**, prescreve um link de dados protocolo de segurança de nível para evitar que um nó sem fio leia ou interfira com mensagens enviadas entre outro par de nós sem fio. Também é conhecido como nome comercial **WPA2 (WiFi Protected Access 2)**. WPA simples é um esquema provisório que implementa um subconjunto de 802.11i. Deve ser evitado em favor do WPA2.

Descreveremos o 802.11i em breve, mas primeiro observaremos que é uma substituição para **WEP (Wired Equivalent Privacy)**, a primeira geração do protocolo de segurança 802.11 cols. WEP foi projetado por um comitê de padrões de rede, que é uma empresa

processo completamente diferente do que, por exemplo, a forma como o NIST selecionou o design de

AES. Os resultados foram devastadores. O que há de errado com isso? Quase todos coisas do ponto de vista da segurança. Por exemplo, WEP criptografado dados para confidencialidade por XORing-los com a saída de uma cifra de fluxo. Infelizmente, arranjos de keying fracos significavam que a saída era frequentemente reutilizada. Isto levou a maneiras triviais de derrotá-lo. Como outro exemplo, a verificação de integridade foi baseada

em um CRC de 32 bits. Esse é um código eficiente para detectar erros de transmissão, mas não é um mecanismo criptograficamente forte para derrotar invasores.

Essas e outras falhas de design tornaram o WEP muito fácil de comprometer. O primeiro demonstração prática de que o WEP estava quebrado veio quando Adam Stubblefield estava estagiário na AT&T (Stubblefield et al., 2002). Ele foi capaz de codificar e testar um ataque descrito por Fluhrer et al. (2001) em uma semana, na maioria das vezes foi gasto para convencer a gerência a comprar para ele um cartão WiFi para usar em sua experiência. Software para quebrar senhas WEP em um minuto agora está disponível gratuitamente e o uso de WEP é fortemente desencorajado. Embora evite

O acesso não fornece nenhuma forma real de segurança. O grupo 802.11i foi colocado juntos com pressa quando ficou claro que o WEP estava seriamente quebrado. Produziu um padrão formal em junho de 2004.

Agora vamos descrever 802.11i, que fornece segurança real se for configurado e usado corretamente. Existem dois cenários comuns em que o WPA2 é usado. O primeiro é uma configuração corporativa, em que uma empresa tem um serviço de autenticação separado

ver que tem um banco de dados de nome de usuário e senha que pode ser usado para determinar se um

o cliente sem fio tem permissão para acessar a rede. Nesta configuração, os clientes usam standartos padrão para se autenticarem na rede. Os principais padrões são **802.1X**, com o qual o ponto de acesso permite ao cliente manter um diálogo com o servidor de autenticação e observa o resultado, e **EAP (Extensible Authentication protocol de autenticação)** (RFC 3748), que informa como o cliente e o serviço de autenticação interagir. Na verdade, o EAP é uma estrutura e outros padrões definem o protocolo mensagens. No entanto, não vamos nos aprofundar nos muitos detalhes dessa troca porque eles não importam muito para uma visão geral.

O segundo cenário é em um ambiente doméstico no qual não há autenticação servidor. Em vez disso, há uma única senha compartilhada que é usada pelos clientes para acessar a rede sem fio. Esta configuração é menos complexa do que ter uma autenticação servidor, é por isso que é usado em casa e em pequenas empresas, mas é menos seguro também. A principal diferença é que com um servidor de autenticação cada cliente obtém uma chave para criptografar o tráfego que não é conhecido pelos outros clientes. Com uma única senha compartilhada, diferentes chaves são derivadas para cada cliente, mas todos os clientes têm a mesma senha e podem derivar as chaves uns dos outros, se quiserem.

As chaves usadas para criptografar o tráfego são calculadas como parte de um handshake de autenticação. O aperto de mão acontece logo após o cliente se associar com uma rede sem fio e autentica com um servidor de autenticação, se houver

1. No início do handshake, o cliente tem a senha de rede compartilhada palavra ou sua senha para o servidor de autenticação. Esta senha é usada para derivar uma chave mestra. No entanto, a chave mestra não é usada diretamente para criptografar pacotes. Isto

é uma prática criptográfica padrão para derivar uma chave de sessão para cada período de uso, para mudar a chave para diferentes sessões, e expor a chave mestra para observar tão pouco quanto possível. É essa chave de sessão que é calculada no handshake.

A chave de sessão é calculada com o handshake de quatro pacotes mostrado na Fig. 8-31. Primeiro, o AP (ponto de acesso) envia um número aleatório para identificação. Correúnneros dom usados apenas uma vez em protocolos de segurança como este são chamados de **nonces**,

o que é mais ou menos uma contração de "número usado uma vez." O cliente também escolhe seu próprio nonce. Ele usa os nonces, seu endereço MAC e do AP, e o dominar chave para calcular uma chave de sessão, K_s . A chave de sessão é dividida em porções, cada um dos quais é usado para finalidades diferentes, mas omitimos esse detalhe. Agora o cliente tem chaves de sessão, mas o AP não. Assim, o cliente envia seu nonce para o AP, e o AP executa o mesmo cálculo para derivar a mesma sessão chaves. Os nonces podem ser enviados em aberto porque as chaves não podem ser derivadas de sem informações secretas extras. A mensagem do cliente está protegida

SEC. 8,6
SEGURANÇA DE COMUNICAÇÃO

825

com uma verificação de integridade chamada **MIC (Message Integrity Check)** com base na chave de sessão. O AP pode verificar se o MIC está correto e, portanto, a mensagem de fato deve ter vindo do cliente, após calcular as chaves de sessão. Um MIC é apenas outro nome para um código de autenticação de mensagem, como em um HMAC. O termo MIC é frequentemente usado em vez de protocolos de rede devido ao potencial de confusão com endereços MAC (Medium Access Control).

Cliente
Nonce AP
Nonce c , MIC s
K s (K G), MIC s
2
4
1
3
Acesso
P
unta
(AP)

```
Sessão de computação
chaves K s do MAC
endereços, nonces,
e chave mestra
Distribuir chave de grupo, K G
Verificar
cliente
tem K s
Verificar
AP
tem K s
Reconhecer
Sessão de computação
chaves K s , mesmo
como o cliente
K s (ACK), MIC s
```

Figura 8-31. O handshake de configuração de chave 802.11i.

Nas duas últimas mensagens, o AP distribui uma chave de grupo, K_G , para o cliente, e o cliente confirma a mensagem. O recebimento dessas mensagens permite ao cliente verifique se o AP possui as chaves de sessão corretas e vice-versa. A chave do grupo é usado para tráfego de broadcast e multicast na LAN 802.11. Porque o resultado de o aperto de mão é que cada cliente tem suas próprias chaves de criptografia, nenhuma dessas chaves pode ser usado pelo AP para transmitir pacotes a todos os clientes sem fio; um separado a cópia da taxa precisaria ser enviada a cada cliente usando sua chave. Em vez disso, uma chave compartilhada

é distribuído para que o tráfego de transmissão possa ser enviado apenas uma vez e recebido por todos

os clientes. Deve ser atualizado à medida que os clientes saem e ingressam na rede.

Finalmente, chegamos à parte em que as chaves são realmente usadas para fornecer segurança ty. Dois protocolos podem ser usados em 802.11i para fornecer confidencialidade de mensagem, em tegridade e autenticação. Como o WPA, um dos protocolos, chamado **TKIP (Temporar Key Integrity Protocol)**, foi uma solução provisória. Foi projetado para im- provar a segurança em cartões 802.11 antigos e lentos, de modo que pelo menos alguma segurança que seja

melhor do que WEP pode ser implementado como uma atualização de firmware. No entanto, também

agora foi quebrado, então você está melhor com o outro protocolo recomendado,

CCMP. O que significa CCMP? É a abreviação de algo espetacular

Nome Modo de contador com encadeamento de bloco de criptografia Código de autenticação de mensagem Pro-

tocol. Vamos chamá-lo apenas de CCMP. Você pode chamá-lo do que quiser.

Página 850

826

SEGURANÇA DE REDE

INDIVÍDUO. 8

O CCMP funciona de maneira bastante direta. Ele usa criptografia AES com um Chave de 128 bits e tamanho do bloco. A chave vem da chave de sessão. Para fornecer con fidencialidade, as mensagens são criptografadas com AES no modo de contador. Lembre-se de que nós

discutido modos de cifra na seção 8.2.3. Esses modos são o que evita o mesmo mensagem seja criptografada para o mesmo conjunto de bits todas as vezes. Modo contador mistura um contador na criptografia. Para fornecer integridade, a mensagem, incluindo campos de cabeçalho, é criptografado com o modo de encadeamento de bloco de cifra e o último de 128 bits

bloco é mantido como o MIC. Em seguida, tanto a mensagem (criptografada com modo de contador) e o MIC são enviados. O cliente e o AP podem, cada um, executar essa criptografia ou verifique essa criptografia quando um pacote sem fio for recebido. Para transmissão ou multi- lançar mensagens, o mesmo procedimento é usado com a chave de grupo.

Segurança Bluetooth

Bluetooth tem um alcance consideravelmente mais curto do que 802.11, então não pode ser facilmente

atacado do estacionamento, mas a segurança ainda é um problema aqui. Por exemplo, imagine que o computador de Alice está equipado com um teclado Bluetooth sem fio. No a ausência de segurança, se Trudy por acaso estivesse no escritório ao lado, ela poderia ler tudo que Alice digitou, incluindo todos os e-mails enviados. Ela também poderia capturar tudo o que o computador de Alice enviou para a impressora Bluetooth ao lado dele (por exemplo, e-mail recebido e relatórios confidenciais). Felizmente, o Bluetooth tem uma esquema de segurança de borato para tentar frustrar os Trudies do mundo. Vamos agora resumir as principais características dele.

Bluetooth versão 2.1 e posterior tem quatro modos de segurança, variando de nada em tudo para criptografia total de dados e controle de integridade. Tal como acontece com 802.11, se a segurança for desativada (o padrão para dispositivos mais antigos), não há segurança. A maioria dos usuários tem curiosidade desligada até que uma violação grave ocorra; então eles ligam. No mundo agrícola, essa abordagem é conhecida como trancar a porta do celeiro atrás do cavalo escapou.

O Bluetooth oferece segurança em várias camadas. Na camada física, frequency hopping oferece um pouco de segurança, mas como qualquer dispositivo Bluetooth que se move para uma piconet deve ser informada a sequência de salto de frequência, consequência obviamente não é um segredo. A verdadeira segurança começa quando o recém-chegado

escravo pede um canal com o mestre. Antes do Bluetooth 2.1, dois dispositivos eram assumidos para compartilhar uma chave secreta configurada com antecedência. Em alguns casos, ambos são

conectados pelo fabricante (por exemplo, para um fone de ouvido e telefone celular vendidos como um
unidade). Em outros casos, um dispositivo (por exemplo, o fone de ouvido) tem uma chave com fio e

o usuário tem que inserir essa chave em outro dispositivo (por exemplo, o telefone celular) como um número decimal. Essas chaves compartilhadas são chamadas de **chaves de acesso**. Infelizmente, as chaves de acesso são frequentemente codificadas para "1234" ou outro valor previsível, e em qualquer

caso são quatro dígitos decimais, permitindo apenas 10⁴ escolhas. Com par seguros simples no Bluetooth 2.1, os dispositivos escolhem um código em um intervalo de seis dígitos, o que torna a senha muito menos previsível, mas ainda longe de ser segura.

Página 851

SEC. 8,6
SEGURANÇA DE COMUNICAÇÃO

827

Para estabelecer um canal, o escravo e o mestre verificam se o outro conhece a chave de acesso. Nesse caso, eles negociam se o canal será criptografado, integridade controlada, ou ambos. Em seguida, eles selecionam uma chave de sessão aleatória de 128 bits, alguns

de cujos bits podem ser públicos. O ponto de permitir este enfraquecimento chave é com com restrições governamentais em vários países, destinadas a prevenir a exportação ou uso de chaves por mais tempo do que o governo pode quebrar.

A criptografia usa uma cifra de fluxo chamada E₀; o controle de integridade usa SAFER+. Ambos são cifras de bloco de chave simétrica tradicionais. SAFER+ foi submetido ao AES bake-off, mas foi eliminado na primeira rodada porque foi mais lento que os outros candidatos. O Bluetooth foi finalizado antes da cifra AES ser escolhida; caso contrário, provavelmente teria usado Rijndael.

A criptografia real usando a cifra de fluxo é mostrada na Figura 8-14, com o texto simples XORed com o fluxo de chaves para gerar o texto cifrado. Infelizmente, o próprio E₀ (como RC4) pode ter fraquezas fatais (Jakobsson e Wetzel, 2001).

Embora não tenha sido quebrado no momento em que este livro foi escrito, suas semelhanças com o A5 / 1

cifra, cuja falha espétacular compromete todo o tráfego de telefone GSM, são

motivo de preocupação (Biryukov et al., 2000). Às vezes surpreende as pessoas (incluindo os autores deste livro), que no perene jogo de gato e rato entre os criptógrafos e os criptoanalistas, os criptanalistas estão tão frequentemente em jogo lado ning.

Outro problema de segurança é que o Bluetooth autentica apenas dispositivos, não usuários, portanto, o roubo de um dispositivo Bluetooth pode dar ao ladrão acesso às informações financeiras e outras contas. No entanto, o Bluetooth também implementa segurança nas camadas superiores, então, mesmo no caso de uma violação de segurança no nível do link, alguma segurança pode permanecer, especialmente para aplicativos que requerem um código PIN a ser inserido manualmente a partir de algum tipo de teclado para completar a transação.

8.7 PROTOCOLOS DE AUTENTICAÇÃO

A **autenticação** é a técnica pela qual um processo verifica se sua comunicação o parceiro de cação é quem deve ser e não um impostor. Verificando a identidade de um processo remoto diante de um intruso ativo e malicioso é surpreendentemente difícil e requer protocolos complexos baseados em criptografia. Nesta secção, vamos estudar alguns dos muitos protocolos de autenticação que são usados em inseguros redes de computadores.

À parte, algumas pessoas confundem autorização com autenticação.

A autenticação trata da questão de saber se você está realmente se comunicando com um processo específico. A autorização se preocupa com o que esse processo realiza pretendido fazer. Por exemplo, digamos que um processo do cliente conte um servidor de arquivos e diga: " I

sou o processo de Scott e quero deletar o arquivo *cookbook.old* ." Do arquivo ser- ponto de vista de ver, duas questões devem ser respondidas:

Página 852

828

SEGURANÇA DE REDE
INDIVÍDUO. 8

1. Este é realmente o processo de Scott (autenticação)?

2. Scott tem permissão para excluir *cookbook.old* (autorização)?

Só depois de ambas as questões terem sido respondidas inequivocamente afirmativamente mative pode a ação solicitada ocorrer. A primeira pergunta é realmente a chave

1. Uma vez que o servidor de arquivos sabe com quem está falando, a verificação da autorização é apenas uma questão de procurar entradas em tabelas ou bancos de dados locais. Por este motivo, nós irá se concentrar na autenticação nesta seção.

O modelo geral que basicamente todos os protocolos de autenticação usam é este.

Alice começa enviando uma mensagem para Bob ou para um **KDC** confiável (**Key Distribuição**), que se espera seja honesto. Várias outras mensagens ex- as mudanças seguem em várias direções. Enquanto essas mensagens são enviadas, Trudy pode interceptar, modificar ou reproduzi-los a fim de enganar Alice e Bob ou apenas para atrapalhar as obras.

No entanto, quando o protocolo for concluído, Alice tem certeza de que ela está falando falar com Bob e Bob tem certeza de que ele está falando com Alice. Além disso, na maioria das tocols, os dois também terão estabelecido uma **chave de sessão** secreta para uso em a próxima conversa. Na prática, por motivos de desempenho, todo o tráfego de dados é criptografado usando criptografia de chave simétrica (normalmente AES ou DES triplo), al- embora a criptografia de chave pública seja amplamente usada para os protocolos de autenticação eles próprios e para estabelecer a chave de sessão.

O objetivo de usar uma nova chave de sessão escolhida aleatoriamente para cada nova con- conexão é minimizar a quantidade de tráfego que é enviada com o segredo dos usuários chaves ou chaves públicas, para reduzir a quantidade de texto cifrado que um intruso pode obter, e para minimizar o dano causado se um processo falhar e seu dump principal cair no mãos erradas. Esperançosamente, a única chave presente será a chave de sessão. Todos as chaves permanentes deveriam ter sido cuidadosamente zeradas depois que a sessão foi

estabelecido.

8.7.1 Autenticação baseada em uma chave secreta compartilhada

Para o nosso primeiro protocolo de autenticação, vamos supor que Alice e Bob al-pronto compartilhe uma chave secreta, K_{AB} . Esta chave compartilhada pode ter sido combinada ao telefone ou pessoalmente, mas, em qualquer caso, não na (insegura) rede.

Este protocolo é baseado em um princípio encontrado em muitos protocolos de autenticação: uma parte envia um número aleatório para a outra, que então o transforma em um número especial forma e retorna o resultado. Esses protocolos são chamados **de proto -desafio-resposta** cols. Neste e nos protocolos de autenticação subsequentes, a seguinte notação irá ser usado:

A, B são as identidades de Alice e Bob.

R_i 's são os desafios, onde i identifico o desafiante.

K_i 's são chaves, onde i indica o proprietário.

K_s é a chave de sessão.

Página 853

SEC. 8,7

PROTÓCOLOS DE AUTENTICAÇÃO

829

A sequência de mensagens para o nosso primeiro protocolo de autenticação de chave compartilhada é ilustrada

tratado na Fig. 8-32. Na mensagem 1, Alice envia sua identidade, A , para Bob de uma forma que Bob entende. Bob, é claro, não tem como saber se esta mensagem veio de Alice ou de Trudy, então ele escolhe um desafio, um grande número aleatório ber, R_B , e envia de volta para "Alice" como mensagem 2, em texto simples. Alice então en-criptografa a mensagem com a chave que ela compartilha com Bob e envia o texto cifrado, $K_{AB}(R_B)$, de volta à mensagem 3. Quando Bob vê esta mensagem, ele imediatamente sabe que veio de Alice porque Trudy não conhece K_{AB} e, portanto, poderia não o gerou. Além disso, uma vez que R_B foi escolhido aleatoriamente de um grande espaço (digamos, números aleatórios de 128 bits), é muito improvável que Trudy tivesse visto R_B e sua resposta em uma sessão anterior. É igualmente improvável que ela pudesse adivinhe a resposta correta a qualquer desafio.

UMA
Alice
 R_B
1
2
4
5
3
 $K_{AB}(R_B)$
 $K_{AB}(R_A)$
Prumo
 R_A

Figura 8-32. Autenticação bidirecional usando um protocolo de resposta de desafio.

Neste ponto, Bob tem certeza de que está falando com Alice, mas Alice não tem certeza de nada coisa. Pelo que Alice sabe, Trudy pode ter interceptado a mensagem 1 e enviado voltar R_B em resposta. Talvez Bob tenha morrido ontem à noite. Para descobrir com quem ela está falando

ing, Alice escolhe um número aleatório, R_A , e o envia para Bob como texto simples, em mensagem sábio 4. Quando Bob responde com $K_{AB}(R_A)$, Alice sabe que ela está falando com Bob. E se eles desejam estabelecer uma chave de sessão agora, Alice pode escolher uma, K_s , e enviá-la para Bob criptografado com K_{AB} .

O protocolo da Figura 8-32 contém cinco mensagens. Vamos ver se podemos ser inteligente e elimine alguns deles. Uma abordagem é ilustrada na Fig. 8-33. Aqui Alice inicia o protocolo de desafio-resposta em vez de esperar que Bob o faça.

Da mesma forma, enquanto responde ao desafio de Alice, Bob envia o seu próprio. o o protocolo inteiro pode ser reduzido a três mensagens em vez de cinco.

Este novo protocolo é uma melhoria em relação ao original? Em certo sentido, é: é mais curto. Infelizmente, também está errado. Sob certas circunstâncias, Trudy pode derrotar este protocolo usando o que é conhecido como um **ataque de reflexão**. Em particular

ular, Trudy pode quebrá-lo se for possível abrir várias sessões com Bob em uma vez. Esta situação seria verdadeira, por exemplo, se Bob fosse um banco e estivesse preparado aceitar várias conexões simultâneas de caixas eletrônicos ao mesmo tempo.

Página 854

830

SEGURANÇA DE REDE

INDIVÍDUO. 8

Alice
1
3
2
R_B, K_{AB}(R_A)
K_{AB}(R_B)
A, R_A
Prumo

Figura 8-33. Um protocolo de autenticação bidirecional reduzido.

O ataque de reflexão de Trudy é mostrado na Fig. 8-34. Tudo começa com a afirmação de Trudy-
ing ela é Alice e enviar R_T . Bob responde, como de costume, com seu próprio desafio,
 R_B . Agora Trudy está presa. O que é que ela pode fazer? Ela não conhece K_{AB} (R_B).

Trudy
1
5
2
R_B, K_{AB}(R_T)
K_{AB}(R_B)
A, R_T
3
4
R_{B2}, K_{AB}(R_B)
A, R_B
Primeira sessão
Segunda sessão
Primeira sessão
Prumo

Figura 8-34. O ataque de reflexão.

Ela pode abrir uma segunda sessão com a mensagem 3, fornecendo o R_B retirado de
a mensagem 2 como seu desafio. Bob o criptografa calmamente e envia de volta K_{AB} (R_B) em
mensagem 4. Nós sombreamos as mensagens na segunda sessão para torná-las
se destacarem. Agora Trudy tem as informações que faltam, então ela pode preencher o primeiro
sessão e abortar a segunda. Bob agora está convencido de que Trudy é Alice, então
quando ela pede o saldo de sua conta bancária, ele dá a ela sem questionar.

Então, quando ela pede a ele para transferir tudo para uma conta bancária secreta na Suíça,
ele o faz sem um momento de hesitação.

A moral desta história é:

Projetar um protocolo de autenticação correto é muito mais difícil do que parece.

As quatro regras gerais a seguir frequentemente ajudam o designer a evitar armadilhas comuns:

Página 855

SEC. 8.7

PROTOCOLOS DE AUTENTICAÇÃO

831

1. Faça com que o iniciador prove quem ele é antes de o respondente precisar. Isto
evita que Bob forneça informações valiosas antes que Trudy precise
dê qualquer evidência de quem ela é.

2. Faça com que o iniciador e o respondente usem chaves diferentes para prova, mesmo que
isso significa ter duas chaves compartilhadas, K_{AB} e K'_{AB} .

3. Faça com que o iniciador e o respondente desenhem seus desafios de diferentes
conjuntos. Por exemplo, o iniciador deve usar números pares e os
sponder deve usar números ímpares.

4. Torne o protocolo resistente a ataques envolvendo um segundo paralelo
sessão em que as informações obtidas em uma sessão são usadas em uma
um diferente.

Mesmo que uma dessas regras seja violada, o protocolo pode ser freqüentemente quebrado. Aqui,

todas as quatro regras foram violadas, com consequências desastrosas.

Agora, vamos dar uma olhada mais de perto na Fig. 8-32. Certamente esse protocolo não é subjetivo a um ataque de reflexão? Talvez. É muito sutil. Trudy foi capaz de derrotar nosso protocolo usando um ataque de reflexão porque foi possível abrir um segundo sessão com Bob e induzi-lo a responder às suas próprias perguntas. O que seria aconteceria se Alice fosse um computador de uso geral que também aceitasse vários sessões, em vez de uma pessoa em um computador? Vamos dar uma olhada no que Trudy pode fazer.

Para ver como funciona o ataque de Trudy, consulte a Figura 8-35. Alice começa anunciando identificando sua identidade na mensagem 1. Trudy intercepta esta mensagem e começa a sua própria

sessão com a mensagem 2, alegando ser Bob. Novamente sombreamos a sessão 2 mensagens. Alice responde à mensagem 2 dizendo na mensagem 3: "Você afirma ser Prumo? Prove." Neste ponto, Trudy está presa porque não pode provar que é Bob. O que Trudy faz agora? Ela volta para a primeira sessão, onde é ela

vire para enviar um desafio, e mande o R_A que ela recebeu na mensagem 3. Alice gentilmente responde a ele na mensagem 5, fornecendo assim a Trudy as informações de que ela precisa para enviar mensagem 6 na sessão 2. Neste ponto, Trudy está basicamente livre para casa porque ela respondeu com sucesso ao desafio de Alice na sessão 2. Ela pode agora cancele a sessão 1, envie qualquer número antigo para o resto da sessão 2, e ela terá uma sessão autenticada com Alice na sessão 2.

Mas Trudy é desagradável, e ela realmente quer esfregar isso. Em vez de enviar qualquer número antigo acabou para completar a sessão 2, ela espera até que Alice envie a mensagem 7, O desafio de Alice para a sessão 1. Claro, Trudy não sabe como responder, então ela usa o ataque de reflexão novamente, enviando de volta R_{A2} como mensagem 8. Alice criptografa convenientemente R_{A2} na mensagem 9. Trudy agora volta para a sessão 1 e envia a Alice o número que ela deseja na mensagem 10, convenientemente copiado de o que Alice enviou na mensagem 9. Neste ponto, Trudy tem duas sessões totalmente autenticadas reuniões com Alice.

Este ataque tem um resultado um pouco diferente do ataque aos três-mes-protocolo sábio que vimos na Fig. 8-34. Desta vez, Trudy tem dois autenticados

Página 856

832

SEGURANÇA DE REDE

INDIVÍDUO. 8

UMA

Alice

B

1

2

4

5

3

$K_{AB}(R_A)$

Trudy

R_A

R_A

6

$K_{AB}(R_A)$

7

R_{A2}

8

9

$K_{AB}(R_{A2})$

R_{A2}

10

$K_{AB}(R_{A2})$

Primeira sessão

Primeira sessão

Primeira sessão

Primeira sessão

Segunda sessão

Segunda sessão

Segunda sessão

Figura 8-35. Um ataque de reflexão no protocolo da Fig. 8-32.

conexões com Alice. No exemplo anterior, ela tinha um con autenticado conexão com Bob. Novamente aqui, se tivéssemos aplicado toda a autenticação geral

regras de protocolo discutidas anteriormente, este ataque poderia ter sido interrompido. Para um debate detalhado desses tipos de ataques e como evitá-los, consulte Bird et al. (1993). Eles também mostram como é possível construir protocolos sistematicamente que são comprovadamente corretos. O protocolo mais simples é, no entanto, um pouco complicado, então agora mostraremos uma classe diferente de protocolo que também funciona. O novo protocolo de autenticação é mostrado na Figura 8-36 (Bird et al., 1993). Isto usa um HMAC do tipo que vimos ao estudar o IPsec. Alice começa enviando Bob um nonce, R_A , como mensagem 1. Bob responde selecionando seu próprio nonce, R_B , e enviá-lo de volta junto com um HMAC. O HMAC é formado por edifício uma estrutura de dados que consiste no nonce de Alice, no nonce de Bob, suas identidades e a chave secreta compartilhada, K_{AB} . Esta estrutura de dados é então inserida no HMAC, por exemplo, usando SHA-1. Quando Alice recebe a mensagem 2, ela agora tem R_A (que ela escolheu a si mesma), R_B , que chega como texto simples, as duas identidades e a chave secreta, K_{AB} , que ela sempre conheceu, para poder calcular o HMAC dela própria. Se concordar com o HMAC na mensagem, ela sabe que está falando com Bob porque Trudy não conhece K_{AB} e, portanto, não consegue descobrir qual HMAC enviar. Alice responde a Bob com um HMAC contendo apenas os dois nonces. Trudy pode de alguma forma subverter este protocolo? Não, porque ela não pode forçar essa outra parte para criptografar ou hash um valor de sua escolha, como aconteceu na Fig. 8-34 e Fig. 8-35. Ambos os HMACs incluem valores escolhidos pelo remetente, algo que Trudy não pode controlar.

Página 857

SEC. 8,7
PROTOCOLOS DE AUTENTICAÇÃO

8.33

```

Alice
1
3
2
R_A
Prumo
R_B, HMAC (R_A, R_B, A, B, K_AB )
HMAC (R_A, R_B, K_AB )

```

Figura 8-36. Autenticação usando HMACs.

Usar HMACs não é a única maneira de usar essa ideia. Um esquema alternativo que é frequentemente usado em vez de computar o HMAC em uma série de itens é para encriptografar os itens sequencialmente usando o encadeamento de blocos de criptografia.

8.7.2 Estabelecendo uma chave compartilhada: a troca de chaves Diffie-Hellman

Até agora, presumimos que Alice e Bob compartilham uma chave secreta. Suponha que eles não (porque até agora não existe uma PKI universalmente aceita para assinatura e distribuição de certificados). Como eles podem estabelecer um? Uma maneira seria por Alice ligar para Bob e lhe desse a chave no telefone, mas ele provavelmente começaria dizendo: "Como eu sei que você é Alice e não Trudy?" Eles poderiam tentar marcar um encontro, com cada um trazendo passaporte, carteira de motorista e três principais cartões de crédito, mas sendo pessoas ocupadas, eles podem não ser capazes de encontrar um mutuo

data aceitável para meses. Felizmente, por incrível que pareça, há uma maneira de estranhos estabelecerem uma chave secreta compartilhada em plena luz do dia, até mesmo

com Trudy registrando cuidadosamente cada mensagem.

O protocolo que permite que estranhos estabeleçam uma chave secreta compartilhada é chamado de **Troca de chaves Diffie-Hellman** (Diffie e Hellman, 1976) e funciona da seguinte maneira.

Alice e Bob têm que concordar em dois números grandes, n e g , onde n é primo, $(n - 1)/2$ também é primo e certas condições se aplicam a g . Esses números podem ser público, então qualquer um deles pode simplesmente pegar n e g e dizer ao outro abertamente.

Agora, Alice escolhe um número grande (digamos, 1024 bits), x , e o mantém em segredo. Similarmente,

Bob escolhe um grande número secreto, y .

Alice inicia o protocolo de troca de chaves enviando a Bob uma mensagem con-

taining ($n, g, g^x \bmod n$), como mostrado na Fig. 8-37. Bob responde enviando um mensagem contendo $g^y \bmod n$. Agora Alice aumenta o número que Bob a enviou para o x é o módulo de potência n para obter $(g^y \bmod n)^x \bmod n$. Bob realiza uma operação semelhante para obter $(g^x \bmod n)^y \bmod n$. Pelas leis da aritmética modular, ambos os cálculos rendimento $g^{xy} \bmod n$. E eis que, como num passe de mágica, Alice e Bob de repente compartilham um chave secreta, $g^{xy} \bmod n$.

Página 858

834

SEGURANÇA DE REDE INDIVÍDUO. 8

```

1
Alice
escolhe x
Prumo
escolhe y
2
g^y mod n
n, g, g^x mod n
Alice calcula
(g^y mod n)^x
= g^{xy} mod n
Bob calcula
(g^x mod n)^y
= g^{xy} mod n
Prumo
Alice
mod n
mod n

```

Figura 8-37. A troca de chaves Diffie-Hellman.

Trudy, é claro, viu as duas mensagens. Ela conhece g e n da mensagem

1. Se ela pudesse calcular x e y , ela poderia descobrir a chave secreta. O problema é, dado apenas $g^x \bmod n$, ela não pode encontrar x . Nenhum algoritmo prático para computação módulo de logaritmos discretos um número primo muito grande é conhecido.

Para tornar este exemplo mais concreto, usaremos o (completamente irrealista) valores de $n = 47$ e $g = 3$. Alice escolhe $x = 8$ e Bob escolhe $y = 10$. Ambos estes são mantidos em segredo. A mensagem de Alice para Bob é $(47, 3, 28)$ porque $3^8 \bmod 47$ é 28. A mensagem de Bob para Alice é (17) . Alice calcula $17^8 \bmod 47$, que é 4. Bob calcula $28^{10} \bmod 47$, que é 4. Alice e Bob agora determinaram independentemente descobriu que a chave secreta agora é 4. Para encontrar a chave, Trudy agora tem que resolver a equação $3^x \bmod 47 = 28$, que pode ser feita por pesquisa exaustiva por pequenos números fibras como esta, mas não quando todos os números têm centenas de bits. Todos atualmente algoritmos conhecidos simplesmente levam muito tempo, mesmo em massivamente paralelos, supercomputadores rápidos.

Apesar da elegância do algoritmo Diffie-Hellman, há um problema:

quando Bob recebe o triplo $(47, 3, 28)$, como ele sabe que é de Alice e não de Trudy? Não há como ele saber. Infelizmente, Trudy pode explorar isso fato para enganar Alice e Bob, conforme ilustrado na Figura 8-38. Aqui, enquanto Alice e Bob estão escolhendo x e y , respectivamente, Trudy escolhe seu próprio número aleatório, z . Alice envia a mensagem 1, destinada a Bob. Trudy o intercepta e envia mensagens sábio 2 para Bob, usando o g e n corretos (que são públicos de qualquer maneira), mas com ela próprio z em vez de x . Ela também envia a mensagem 3 de volta para Alice. Depois Bob envia mensagem 4 para Alice, que Trudy novamente intercepta e guarda.

Agora todo mundo faz a aritmética modular. Alice calcula a chave secreta como $g^{xz} \bmod n$, e Trudy também (para mensagens para Alice). Bob calcula $g^{yz} \bmod n$ e Trudy também (para mensagens para Bob). Alice acha que ela está falando com Bob, então ela estabelece uma chave de sessão (com Trudy). Bob também. Cada mensagem que Alice envia na sessão criptografada é capturado por Trudy, armazenado, modificado se desejado e (opcionalmente) passado para Bob. Da mesma forma, na outra direção, Trudy vê tudo e pode modificar todas as mensagens à vontade, enquanto Alice e Bob tem a ilusão de que eles têm um canal seguro entre eles. Por esta

SEC. 8,7
PROTOCOLOS DE AUTENTICAÇÃO

835

```

1
Alice
escolhe x
Trudy
escolhe z
3
gz mod n
n, g, gx mod n
Trudy
2
Prumo
escolhe y
4
gy mod n
n, g, gz mod n
Prumo
Alice

```

Figura 8-38. O ataque man-in-the-middle.

razão, o ataque é conhecido como ataque **man-in-the-middle**. Também é chamado de **ataque da brigada de balde**, porque vagamente se assemelha a um bombeiro voluntário dos velhos tempos

departamento passando baldes ao longo da linha do caminhão de bombeiros até o incêndio.

8.7.3 Autenticação usando um centro de distribuição de chaves

Configurar um segredo compartilhado com um estranho quase funcionou, mas não exatamente. Em por outro lado, provavelmente não valia a pena fazer em primeiro lugar (uvas verdes emaderência). Para falar com n pessoas dessa maneira, você precisaria de n chaves. Para pessoas populares,

gerenciamento de chaves se tornaria um fardo real, especialmente se cada chave tivesse que ser armazenado em um cartão com chip de plástico separado.

Uma abordagem diferente é apresentar um centro de distribuição de chaves confiável. Nisso modelo, cada usuário tem uma única chave compartilhada com o KDC. Autenticação e sessão gerenciamento de chaves de seção agora passa pelo KDC. O KDC mais simples conhecido protocolo de autenticação envolvendo duas partes e um KDC confiável é descrito em

Fig. 8-39.

```

1
A, KA(B, KS)
KDC
2
Prumo
Alice
KB(A, KS)

```

Figura 8-39. Uma primeira tentativa de protocolo de autenticação usando um KDC.

A ideia por trás deste protocolo é simples: Alice escolhe uma chave de sessão, K_S , e diz o KDC que ela quer falar com Bob usando K_S . Esta mensagem está criptografada

836

SEGURANÇA DE REDE
INDIVÍDUO. 8

com as ações Alice chaves secretas (apenas) com o KDC, K_A . O KDC descriptografa este mensagem, extraindo a identidade de Bob e a chave de sessão. Em seguida, ele constrói um novo mensagem contendo a identidade de Alice e a chave de sessão e envia esta mensagem para Prumo. Essa criptografia é feita com K_B , a chave secreta que Bob compartilha com o KDC. Quando Bob descriptografa a mensagem, ele descobre que Alice quer falar com ele e qual chave ela deseja usar.

A autenticação aqui é gratuita. O KDC conhece aquela mensagem 1 deve ter vindo de Alice, já que ninguém mais seria capaz de criptografá-lo com a chave secreta de Alice. Da mesma forma, Bob sabe que a mensagem 2 deve ter chegado do KDC, em quem ele confia, já que ninguém mais conhece sua chave secreta.

Infelizmente, este protocolo tem uma falha séria. Trudy precisa de algum dinheiro, então ela descobre algum serviço legítimo que pode prestar a Alice, faz um

oferta atraente e consegue o emprego. Depois de fazer o trabalho, Trudy educadamente voltou a busca Alice para pagar por transferência bancária. Alice então estabelece uma chave de sessão com ela

banqueiro, Bob. Em seguida, ela envia a Bob uma mensagem solicitando que o dinheiro seja transferido para a conta de Trudy.

Enquanto isso, Trudy está de volta aos velhos tempos, bisbilhotando na rede. Ela

copia a mensagem 2 na Figura 8-39 e a solicitação de transferência de dinheiro que a segue.

Mais tarde, ela repassa os dois para Bob, que pensa: "Alice deve ter contratado Trudy novamente. Ela claramente faz um bom trabalho." Bob então transfere uma quantia igual de dinheiro

olhos do relato de Alice para o de Trudy. Algum tempo depois do 50º par de mensagens, Bob corre para fora do escritório para encontrar Trudy para lhe oferecer um grande empréstimo para que ela possa expandir seu

obviamente um negócio de sucesso. Esse problema é chamado de **ataque de repetição**.

Várias soluções para o ataque de repetição são possíveis. O primeiro é incluir um carimbo de data / hora em cada mensagem. Então, se alguém receber uma mensagem obsoleta, pode ser descartado. O problema com esta abordagem é que os relógios nunca são exatamente sincronizados em uma rede, então deve haver algum intervalo durante o qual um timestamp é válido. Trudy pode repetir a mensagem durante este intervalo e obter fora com isso.

A segunda solução é colocar um nonce em cada mensagem. Cada parte então tem que lembre-se de todos os nonces anteriores e rejeite qualquer mensagem que contenha um anterior usado nonce. Mas os nonces têm que ser lembrados para sempre, para que Trudy não tente repetir uma mensagem de 5 anos. Além disso, se alguma máquina travar e perder sua lista de nonce, é novamente vulnerável a um ataque de repetição. Timestamps e nonces podem ser combinados para limitar quanto tempo os nonces devem ser lembrados, mas claramente o protocolo está indo para ficar muito mais complicado.

Uma abordagem mais sofisticada para autenticação mútua é usar um método protocolo de desafio-resposta. Um exemplo bem conhecido de tal protocolo é o Protocolo de **autenticação Needham-Schroeder** (Needham e Schroeder, 1978), uma variante é mostrada na Fig. 8-40.

O protocolo começa com Alice dizendo ao KDC que deseja falar com Bob.

Esta mensagem contém um grande número aleatório, R_A , como um nonce. O KDC envia mensagem de retorno 2 contendo o número aleatório de Alice, uma chave de sessão e um tiquete

Página 861

SEC. 8,7

PROTOCOLOS DE AUTENTICAÇÃO

837

¹
 R_A , A, B

²

$K_A(R_A, B, K_S, K_B(A, K_S))$

KDC

³

Prumo

Alice

$K_B(A, K_S), K_S(R_{A2})$

⁴

$K_S(R_{A2}-1), R_B$

⁵

$K_S(R_B-1)$

Figura 8-40. O protocolo de autenticação Needham-Schroeder.

que ela pode enviar para Bob. O objetivo do número aleatório, R_A , é garantir a Alice essa mensagem 2 é nova, e não uma repetição. A identidade de Bob também está incluída no caso Trudy tem ideias engraçadas sobre como substituir B na mensagem 1 por sua própria identidade de modo que o KDC irá encriptar o bilhete no final da mensagem 2 com K_T em vez de K_B .

O tiquete criptografado com K_B é incluído dentro da mensagem criptografada para evitar Trudy de substituí-lo por outra coisa no caminho de volta para Alice.

Alice agora envia o tiquete para Bob, junto com um novo número aleatório, R_{A2} , en-

crypted com a chave de sessão, K_S . Na mensagem 4, Bob envia de volta $K_S(R_A2 - 1)$ para provar a Alice que ela está falando com o verdadeiro Bob. Enviar de volta $K_S(R_A2)$ seria não funcionou, pois Trudy poderia simplesmente tê-lo roubado da mensagem 3.

Depois de receber a mensagem 4, Alice agora está convencida de que está falando com Bob e que nenhum replay poderia ter sido usado até agora. Afinal, ela acabou de gerar R_A2 alguns milissegundos atrás. O objetivo da mensagem 5 é convencer Bob de que é na verdade, ele está falando com Alice, e nenhum replay está sendo usado aqui também. Tendo cada parte gerar um desafio e responde a um, a possibilidade de qualquer tipo de ataque de repetição é eliminado.

Embora este protocolo pareça bastante sólido, ele tem uma pequena fraqueza. E se Trudy sempre consegue obter uma chave de sessão antiga em texto simples, ela pode iniciar um nova sessão com Bob, reproduzindo a mensagem 3 que corresponde ao chave comprometida e convencê-lo de que ela é Alice (Denning e Sacco, 1981).

Desta vez, ela pode saquear a conta bancária de Alice sem ter que realizar o serviço legítimo, mesmo uma vez.

Needham e Schroeder (1987) publicaram posteriormente um protocolo que corrige este problema. Na mesma edição da mesma revista, Otway e Rees (1987) também publicaram eliminou um protocolo que resolve o problema de forma mais curta. A Figura 8-41 mostra um protocolo de Otway-Rees ligeiramente modificado.

No protocolo Otway-Rees, Alice começa gerando um par de números: R_A , que será usado como um identificador comum, e R_B , que Alice usará usar para desafiar Bob. Quando Bob recebe esta mensagem, ele constrói uma nova mensagem da parte criptografada da mensagem de Alice e uma análoga da sua própria.

Página 862

838

SEGURANÇA DE REDE INDIVÍDUO. 8

4
 $K_A(R_A, K_S)$
3
2
 $K_B(R_B, K_S)$
KDC
1
Prumo
Alice
 $A, B, R, K_A(A, B, R, R_A)$
 $A, K_A(A, B, R, R_A),$
 $B, K_B(A, B, R, R_B)$

Figura 8-41. O protocolo de autenticação Otway-Rees (ligeiramente simplificado).

Ambas as partes criptografadas com K_A e K_B identificam Alice e Bob, contêm o com identificador mon e conter um desafio.

O KDC verifica se o R em ambas as partes é o mesmo. Pode não ser se Trudy adulterou R na mensagem 1 ou substituiu parte da mensagem 2. Se os dois R s coincidem, o KDC acredita que a mensagem de solicitação de Bob é válida. Isso então gera uma chave de sessão e a criptografa duas vezes, uma para Alice e outra para Bob. Cada mensagem contém o número aleatório do receptor, como prova de que o KDC, e não Trudy, gerou a mensagem. Neste ponto, Alice e Bob estão posse da mesma chave de sessão e pode iniciar a comunicação. A primeira vez eles trocam mensagens de dados, cada um pode ver que o outro tem um idêntico cópia do K_S , então a autenticação é concluída.

8.7.4 Autenticação usando Kerberos

Um protocolo de autenticação usado em muitos sistemas reais (incluindo Windows 2000 e versões posteriores) é **Kerberos**, que é baseado em uma variante de Needham-Schroeder. É o nome de um cão com várias cabeças na mitologia grega que costumava vigiar a entrada de Hades (presumivelmente para manter os indesejáveis fora). Kerberos era projetado no MIT para permitir que usuários de estações de trabalho acessem recursos de rede em um maneira segura. Sua maior diferença em relação a Needham-Schroeder é a suposição de que todos os relógios estão bastante bem sincronizados. O protocolo passou por vários

iterações. V5 é aquele que é amplamente utilizado na indústria e definido na RFC 4120. A versão anterior, V4, foi finalmente retirada depois que sérias falhas foram encontradas (Yu et al., 2004). V5 melhora em V4 com muitas pequenas mudanças no protocolo e alguns recursos aprimorados, como o fato de não depender mais do desatualizado DES.

Para obter mais informações, consulte Neuman e Ts'o (1994).

Kerberos envolve três servidores além de Alice (uma estação de trabalho cliente):

1. Servidor de autenticação (AS): Verifica os usuários durante o login.
2. Servidor de concessão de tíquetes (TGS): Emite " comprovante de tíquetes de identidade ".
3. Bob, o servidor: Realmente faz o trabalho que Alice deseja que seja executado.

Página 863

SEC. 8,7

PROTOCOLOS DE AUTENTICAÇÃO

839

AS é semelhante a um KDC no sentido de que compartilha uma senha secreta com cada usuário. O trabalho do TGS é emitir tickets que possam convencer os servidores reais de que o portador de um

O ingresso TGS realmente é quem ele ou ela afirma ser.

Para iniciar uma sessão, Alice se senta em uma estação de trabalho pública arbitrária e digita o nome dela. A estação de trabalho envia seu nome e o nome do TGS para o AS em texto simples, como mostrado na mensagem 1 da Fig. 8-42. O que volta é uma chave de sessão e um bilhete, $K_{TGS}(A, K_s, t)$, destinado ao TGS. A chave de sessão é criptografada usando a chave secreta de Alice, para que apenas Alice possa descriptografá-la. Somente quando mensagem 2

chega a estação de trabalho pede a senha de Alice - não antes disso, o a senha é então usada para gerar K_A , a fim de descriptografar a mensagem 2 e obter a chave de sessão.

Neste ponto, a estação de trabalho sobrescreve a senha de Alice para se certificar de que só fica dentro da estação de trabalho por alguns milissegundos, no máximo. Se Trudy tentar log-in entrando como Alice, a senha que ela digitar estará errada e a estação de trabalho detectar isso porque a parte padrão da mensagem 2 estará incorreta.

Alice
COMO
TGS
Primo
 $K_{AB}(A, t), K_B(A, B, K_{AB}, t)$
 A, TGS
 $K_A(TGS, K_s, t), K_{TGS}(A, K_s, t)$
 $B, K_s(A, t), K_{TGS}(A, K_s, t)$
 $K_s(B, K_{AB}, t), K_B(A, B, K_{AB}, t)$
 $K_{AB}(t)$
6
5
2
4
1
3

Figura 8-42. A operação do Kerberos V5.

Depois de fazer login, Alice pode dizer à estação de trabalho que deseja entrar em contato com Bob o servidor de arquivos. A estação de trabalho então envia a mensagem 3 para o TGS solicitando um bilhete para usar com Bob. O elemento-chave nesta solicitação é o tíquete $K_{TGS}(A, K_s, t)$, que é criptografado com a chave secreta do TGS e usado como prova de que o remetente realmente é Alice. O TGS responde na mensagem 4 criando uma chave de sessão, K_{AB} , para Alice usar com Bob. Duas versões são enviadas de volta. O primeiro é criptografado com apenas K_s , para que Alice possa lê-lo. O segundo é outro tíquete, criptografado com Chave de Bob, K_B , para que Bob possa lê-la.

Página 864

840

SEGURANÇA DE REDE

INDIVÍDUO. 8

Trudy pode copiar a mensagem 3 e tentar usá-la novamente, mas será frustrada pelo

timestamp criptografado, t , enviado junto com ele. Trudy não pode substituir o carimbo de data / hora com um mais recente, porque ela não conhece K_s , a chave de sessão que Alice usa para falar com o TGS. Mesmo que Trudy repita a mensagem 3 rapidamente, tudo o que ela receberá é outra cópia da mensagem 4, que ela não conseguiu decifrar da primeira vez e não vai ser capaz de descriptografar a segunda vez também.

Agora Alice pode enviar K_{AB} para Bob por meio do novo tiquete para estabelecer uma sessão com ele (mensagem 5). Esta troca também é registrada com data e hora. A resposta opcional (mensagem 6) é a prova para Alice de que ela está realmente falando com Bob, não com Trudy. Após esta série de trocas, Alice pode se comunicar com Bob disfarçadamente de K_{AB} . Se mais tarde ela decidir que precisa falar com outro servidor, Carol, ela simplesmente voltará turfas Mensagem 3 para o TGS, só agora especificando C em vez de B . O TGS irá responder prontamente com um tiquete criptografado com K_C que Alice pode enviar para Carol e que Carol aceitará como prova que veio de Alice.

O objetivo de todo esse trabalho é que agora Alice pode acessar servidores em toda a rede trabalhar de forma segura e sua senha nunca terá que passar pela rede. De fato, só precisava ficar em sua própria estação de trabalho por alguns milissegundos. No entanto, observe que cada servidor faça sua própria autorização. Quando Alice apresenta seu ingresso para Bob, isso apenas prova a Bob quem o enviou. Precisamente o que Alice tem permissão para fazer é com Bob.

Já que os designers do Kerberos não esperavam que o mundo inteiro confiasse em um único servidor de autenticação, eles previram ter vários **domínios**, cada um com seu próprio AS e TGS. Para obter um tiquete para um servidor em um reino distante, Alice iria peça a seu próprio TGS um bilhete aceito pelo TGS no reino distante. Se o distant TGS se registrou com o TGS local (da mesma forma que os servidores locais), o TGS local dará a Alice um ingresso válido no TGS distante. Ela pode então fazer negócios nessa lá, como conseguir tiquetes para servidores naquele reino. Observe, no entanto, que para as partes em dois reinos fazerem negócios, cada um deve confiar no TGS do outro. Caso contrário, eles não podem fazer negócios.

8.7.5 Autenticação usando criptografia de chave pública

A autenticação mútua também pode ser feita usando criptografia de chave pública. Para para começar, Alice precisa obter a chave pública de Bob. Se existe uma PKI com um diretório servidor que distribui certificados para chaves públicas, Alice pode solicitar os de Bob, como mostrado na Fig. 8-43 como mensagem 1. A resposta, na mensagem 2, é um certificado X.509 contendo a chave pública de Bob. Quando Alice verifica se a assinatura está correta, ela envia a Bob uma mensagem contendo sua identidade e um nonce.

Quando Bob recebe esta mensagem, ele não tem ideia se é de Alice ou de Trudy, mas ele brinca e pede ao servidor de diretório o público de Alice (mensagem 4), que ele logo recebe (mensagem 5). Ele então envia uma mensagem para Alice 6, contendo de Alice R_{Um} , seu próprio nonce, R_B , e uma chave de sessão proposta, K_s .

1. Dê
mim
E_B

Figura 8-43. Autenticação mútua usando criptografia de chave pública.

Quando Alice recebe a mensagem 6, ela a descriptografa usando sua chave privada. Ela vê R_A nele, o que lhe dá uma sensação de calor por dentro. A mensagem deve ter vindo de Bob, desde Trudy não tem maneira de determinar R_{Uma} . Além disso, deve ser fresca e não uma repetição, uma vez que ela acabou de enviar Bob R_A . Alice concorda com a sessão por enviando de volta a mensagem 7. Quando Bob vê R_B criptografado com a chave de sessão, ele apenas gerado, ele sabe que Alice tem mensagem 6 e verificada R_A . Bob agora é um campista feliz.

O que Trudy pode fazer para tentar subverter esse protocolo? Ela pode fabricar mensagem 3 e engane Bob para sondar Alice, mas Alice verá um R_A que ela não enviou e não irá prosseguir. Trudy não pode forjar a mensagem 7 de volta para Bob porque ela não conhece R_B ou K_S e não pode determiná-los sem o contato privado de Alice chave. Ela está sem sorte.

8.8 SEGURANÇA DE E-MAIL

Quando uma mensagem de e-mail é enviada entre dois sites distantes, geralmente transitam dezenas de máquinas pelo caminho. Qualquer um deles pode ler e gravar a mensagem sage para uso futuro. Na prática, a privacidade é inexistente, apesar do que muitas pessoas pensar. No entanto, muitas pessoas gostariam de enviar e-mails que podem ser lida pelo destinatário pretendido e mais ninguém: nem seu chefe e nem mesmo seu governo. Esse desejo tem estimulado várias pessoas e grupos a aplicarem o princípios criptográficos que estudamos anteriormente para e-mail para produzir e-mail seguro. No nas seções a seguir, estudaremos um sistema de e-mail seguro amplamente usado, PGP e em seguida, mencione brevemente um outro, S / MIME. Para obter informações adicionais sobre seguro e-mail, consulte Kaufman et al. (2002) e Schneier (1995).

Página 866

842

SEGURANÇA DE REDE
INDIVÍDUO. 8

8.8.1 PGP - Boa privacidade

Nosso primeiro exemplo, **PGP** (**Pretty Good Privacy**) é essencialmente uma criação de uma pessoa, Phil Zimmermann (1995a, 1995b). Zimmermann é uma privacidade advogado cujo lema é: " Se a privacidade for proibida, apenas os fora-da-lei terão privacidade cy. " Lançado em 1991, PGP é um pacote completo de segurança de e-mail que fornece privacidade, autenticação, assinaturas digitais e compactação, tudo em um fácil de usar Formato. Além disso, o pacote completo, incluindo todo o código-fonte, é distribuído gratuitamente pela Internet. Pela qualidade, preço (zero) e facilidade disponibilidade em plataformas UNIX , Linux, Windows e Mac OS, é amplamente utilizado hoje.

PGP criptografa os dados usando uma cifra de bloco chamada **IDEA** (**International Data Algoritmo de criptografia**), que usa chaves de 128 bits. Foi criado na Suíça em uma época em que o DES era visto como contaminado e o AES ainda não havia sido inventado. Vigarista-

Cepitualmente, o IDEA é semelhante ao DES e AES: ele mistura os bits em uma série de rodadas, mas os detalhes das funções de mixagem são diferentes de DES e AES. O gerenciamento de chaves usa RSA e a integridade de dados usa MD5, tópicos que já temos pronto discutido.

O PGP também está envolvido em controvérsias desde o dia 1 (Levy, 1993). Estar porque Zimmermann não fez nada para impedir outras pessoas de colocar PGP no Internet, onde pessoas de todo o mundo podem obtê-lo, afirmou o governo dos EUA que Zimmermann violou as leis dos EUA que proíbem a exportação de munições. o A investigação de Zimmermann pelo governo dos EUA durou 5 anos, mas foi acabou caindo, provavelmente por dois motivos. Primeiro, Zimmermann não colocou

O próprio PGP na Internet, então seu advogado alegou que *ele* nunca exportou nenhum coisa (e então há a pequena questão de se criar um site constitui exportação). Em segundo lugar, o governo acabou percebendo que ganhar um julgamento significava convencer um júri de que um site contendo uma privacidade para download programa foi coberto pela lei de tráfico de armas que proíbe a exportação de guerra material como tanques, submarinos, aeronaves militares e armas nucleares. Anos de publicidade negativa provavelmente também não ajudou muito.

À parte, as regras de exportação são bizarras, para dizer o mínimo. O governo considerou colocar código em um site como uma exportação ilegal e assediou Zimmermann sobre isso por 5 anos. Por outro lado, quando alguém publicou o código-fonte PGP completo, em C, como um livro (em uma fonte grande com um checksum cada página para facilitar a digitalização) e depois exportar o livro, tudo bem com o governo porque os livros não são classificados como munições. A espada é mais poderoso do que a caneta, pelo menos para o tio Sam.

Outro problema que o PGP encontrou envolveu violação de patente. A empresa detentora da patente RSA, RSA Security, Inc., alegou que o uso da RSA pelo PGP algoritmo infringiu sua patente, mas esse problema foi resolvido com o início dos lançamentos em 2.6. Além disso, o PGP usa outro algoritmo de criptografia patenteado, IDEA, cujo uso causou alguns problemas no início.

Página 867

SEC. 8,8

SEGURANÇA DE E-MAIL

843

Uma vez que o PGP é de código aberto, várias pessoas e grupos o modificaram e produziu várias versões. Alguns deles foram projetados para contornar as leis de munições, outras se concentraram em evitar o uso de algoritmos patenteados, e ainda outros queriam transformá-lo em um produto comercial de código fechado. Embora as leis de munições tenham sido ligeiramente liberalizadas (caso contrário, os produtos usando AES não seria exportável dos EUA), e a patente RSA expirou em setembro de 2000, o legado de todos esses problemas é que várias incompatibilidades compatíveis de PGP estão em circulação, sob vários nomes. A discussão abaixo se concentra no PGP clássico, que é a versão mais antiga e simples. Outra versão popular, Open PGP, é descrita na RFC 2440. Outra é o GNU Proteção de privacidade.

PGP usa intencionalmente algoritmos criptográficos existentes ao invés de inventar novos. É amplamente baseado em algoritmos que têm resistido extensa revisão e não foram concebidos ou influenciados por qualquer agência governamental que tente enfraquecer os. Para pessoas que não confiam no governo, esta propriedade é uma grande vantagem. PGP suporta compressão de texto, sigilo e assinaturas digitais e também oferece extensas facilidades de gerenciamento de chaves, mas, estranhamente, não possui facilidades de e-mail.

É como um pré-processador que recebe texto simples como entrada e produz uma cifra assinada texto em base64 como saída. Essa saída pode então ser enviada por e-mail, é claro. Algumas implementações chamam um agente de usuário como a etapa final para realmente enviar a mensagem.

Para ver como o PGP funciona, consideremos o exemplo da Figura 8-44. Aqui Alice deseja enviar uma mensagem de texto simples assinada, P , para Bob de maneira segura. Alice tanto e Bob tem chaves RSA privadas (D_A) e públicas (E_A). Vamos supor que cada um conhece a chave pública do outro; cobriremos o gerenciamento de chaves PGP em breve. Alice começa invocando o programa PGP em seu computador. PGP primeiro hashes sua mensagem, P , usando MD5, e então criptografa o hash resultante usando a chave RSA privada, D_A . Quando Bob finalmente recebe a mensagem, ele pode descriptografar o hash com a chave pública de Alice e verifique se o hash está correto. Mesmo se alguém outra pessoa (por exemplo, Trudy) poderia adquirir o hash neste estágio e descriptografá-lo com a de Alice chave pública conhecida, a força do MD5 garante que seria computa-

É inviável produzir outra mensagem com o mesmo hash MD5. O hash criptografado e a mensagem original agora estão concatenados em um single message, $P1$, e compactado usando o programa ZIP, que usa o Ziv-Algoritmo de Lempel (Ziv e Lempel, 1977). Ligue para a saída desta etapa $P1.Z$. Em seguida, o PGP solicita a Alice alguma entrada aleatória. Tanto o conteúdo quanto a velocidade de digitação é usada para gerar uma chave de mensagem IDEA de 128 bits, K_M (chamada de sessão de chave de sessão na literatura PGP, mas este é realmente um nome impróprio, uma vez que não há sion). K_M agora é usado para criptografar $P1.Z$ com IDEA no modo de feedback de cifra. No Adicionalmente, K_M é criptografado com a chave pública de Bob, E_B . Esses dois componentes são em seguida, concatenado e convertido em base64, como discutimos na seção sobre MIME no cap. 7. A mensagem resultante contém apenas letras, dígitos e os símbolos +, / e =, o que significa que pode ser colocado em um corpo RFC 822 e ser esperado para chegar sem modificações.

Página 868

844

SEGURANÇA DE REDE

INDIVÍDUO. 8

```

MD5
RSA
Fecho eclair
IDÉIA
Base
64
RSA
Texto ASCII para
a rede
P1.Z
P
P1
Original
texto simples
mensagem
de Alice
Concatenação de
P e o assinado
hash de P
Concatenação de
P1.Z criptografado
com IDEA e K_M
criptografado com E_B
Alice é particular
Chave RSA, D_A
P1 comprimido
Público de Bob
Chave RSA, E_B
K_M : Chave de mensagem única para IDEA
: Concatenação
K_M

```

Figura 8-44. PGP em operação para envio de mensagem.

Quando Bob recebe a mensagem, ele reverte a codificação base64 e descriptografa a chave IDEA usando sua chave RSA privada. Usando esta chave, ele descriptografa a mensagem para obter $P1.Z$. Depois de descompactá-lo, Bob separa o texto simples do envelope criptografado e descriptografa o hash usando a chave pública de Alice. Se o hash de texto simples

concorda com seu próprio cálculo MD5, ele sabe que P é a mensagem correta e que veio de Alice.

É importante notar que RSA só é usado em dois lugares aqui: para criptografar o Hash MD5 de 128 bits e para criptografar a chave IDEA de 128 bits. Embora RSA seja lento, precisa criptografar apenas 256 bits, não um grande volume de dados. Além disso, todos os 256 bits de texto simples são excessivamente aleatórios, portanto, uma quantidade considerável de trabalho será

exigido da parte de Trudy apenas para determinar se uma chave adivinhada está correta. O peso da criptografia de serviço é feita pelo IDEA, que é muito mais rápido do que o RSA.

Assim, o PGP fornece segurança, compressão e uma assinatura digital e o faz em um maneira muito mais eficiente do que o esquema ilustrado na Fig. 8-19.

PGP suporta quatro comprimentos de chave RSA. Cabe ao usuário selecionar aquele que é o mais apropriado. Os comprimentos são:

1. Casual (384 bits): pode ser quebrado facilmente hoje.
2. Comercial (512 bits): Quebrável por organizações de três letras.
3. Militar (1024 bits): Não pode ser quebrado por ninguém na terra.
4. Alien (2048 bits): Também não pode ser quebrado por ninguém em outros planetas.

Página 869

SEC. 8,8

SEGURANÇA DE E-MAIL

845

Uma vez que RSA é usado apenas para dois pequenos cálculos, todos devem usar alien-chaves de força o tempo todo.

O formato de uma mensagem PGP clássica é mostrado na Figura 8-45. Numerosos outros formatos também estão em uso. A mensagem tem três partes, contendo a chave IDEA, a assinatura e a mensagem, respectivamente. A parte chave contém não apenas o chave, mas também um identificador de chave, uma vez que os usuários podem ter várias chaves.

```
EU IRIA
do
E B
EU IRIA
do
E A
Sig.
hdr
MD5
cerquilha
Msg
hdr
Arquivo
nome
T
Eu
m
e
T
Eu
m
e
T
y
p
e
s
K M
mensagem
Criptografado
por
E B
D A
Compactado, criptografado por IDEA
Base64
Parte de assinatura
mensagem
parte chave
Parte da mensagem
```

Figura 8-45. Uma mensagem PGP.

A parte da assinatura contém um cabeçalho, o que não nos interessa aqui. O cabeçalho é seguido por um carimbo de data / hora, o identificador da chave pública do remetente que

pode ser usado para descriptografar o hash da assinatura, algum tipo de informação que identifica os algoritmos usados (para permitir que MD6 e RSA2 sejam usados quando forem inventados), e o próprio hash criptografado.

A parte da mensagem também contém um cabeçalho, o nome padrão do arquivo a ser usado se o receptor grava o arquivo no disco, um carimbo de data / hora de criação da mensagem e, finalmente, a própria mensagem.

O gerenciamento de chaves tem recebido grande atenção no PGP, pois é o calcanhar de Aquiles de todos os sistemas de segurança. O gerenciamento de chaves funciona da seguinte maneira. Cada

o usuário mantém duas estruturas de dados localmente: um anel de chave privada e uma chave pública

anel. O **anel de chave privada** contém uma ou mais chaves privadas / públicas pessoais pares. A razão para oferecer suporte a vários pares por usuário é permitir que os usuários mudar suas chaves públicas periodicamente ou quando se pensa que uma foi comprometidos, sem invalidar mensagens atualmente em preparação ou em trânsito. Cada par tem um identificador associado a ele para que o remetente da mensagem possa informar o destinatário qual chave pública foi usada para criptografá-lo. Os identificadores de mensagem consistem em os 64 bits de baixa ordem da chave pública. Os próprios usuários são responsáveis por evitando conflitos em seus identificadores de chave pública. As chaves privadas no disco são encriptografados usando uma senha especial (arbitrariamente longa) para protegê-los contra furtos ataques. O **anel de chave pública** contém as chaves públicas dos correspondentes do usuário. Estes são necessários para criptografar as chaves de mensagem associadas a cada mensagem. Cada entrada

Página 870

846

SEGURANÇA DE REDE INDIVÍDUO. 8

no anel de chave pública contém não apenas a chave pública, mas também sua identidade de 64 bits e uma indicação de quanto fortemente o usuário confia na chave.

O problema que está sendo enfrentado aqui é o seguinte. Suponha que as chaves públicas são mantidos em quadros de avisos. Uma maneira de Trudy ler o e-mail secreto de Bob é atacar o quadro de avisos e substituir a chave pública de Bob por uma de sua escolha.

Mais tarde, quando Alice busca a chave supostamente pertencente a Bob, Trudy pode montar um ataque da brigada de baldes a Bob.

Para evitar esses ataques, ou pelo menos minimizar as consequências deles, Alice precisa saber o quanto confiar no item chamado "chave de Bob" em sua chave pública anel. Se ela souber que Bob entregou pessoalmente a ela um CD-ROM contendo a chave, ela pode definir o valor de confiança para o valor mais alto. É este descentralizado, usuário-con abordagem controlada para gerenciamento de chave pública que diferencia o PGP do centralizado Esquemas de PKI.

No entanto, as pessoas às vezes obtêm chaves públicas, consultando um confiável servidor de chaves. Por esta razão, depois que X.509 foi padronizado, PGP apoiou estes certificados, bem como o mecanismo tradicional de anel de chave pública PGP. Todos atuais versões de PGP têm suporte para X.509.

8.8.2 S / MIME

O empreendimento da IETF em segurança de e-mail, chamado **S / MIME** (**Secure / MIME**), é de-inscrito nas RFCs 2632 a 2643. Fornece autenticação, integridade de dados, sigilo e não-repúdio. Também é bastante flexível, suportando uma variedade de algoritmos criptográficos. Não surpreendentemente, dado o nome, S / MIME integra bem com MIME, permitindo que todos os tipos de mensagens sejam protegidos. Uma variedade de novos cabeçalhos MIME são definidos, por exemplo, para armazenar assinaturas digitais. S / MIME não tem uma hierarquia de certificados rígida começando em uma única raiz, que foi um dos problemas políticos que condenaram um sistema anterior chamado PEM (Privacy Enhanced Mail). Em vez disso, os usuários podem ter várias âncoras de confiança. Contanto que um certificado possa ser rastreado até alguma âncora de confiança que o usuário acredita em, é considerado válido. S / MIME usa os algoritmos e protocolos padrão que nós examinamos até agora, portanto, não discutiremos mais isso aqui. Para o de-caudas, consulte as RFCs.

8.9 SEGURANÇA DA WEB

Acabamos de estudar duas áreas importantes onde a segurança é necessária: cátions e e-mail. Você pode pensar nisso como a sopa e o aperitivo. Agora é hora do curso principal: segurança web. A Web é onde a maioria dos Trudies sair hoje em dia e fazer o trabalho sujo. Nas seções a seguir, iremos veja alguns dos problemas e questões relacionados à segurança na web.

SEC. 8,9
SEGURANÇA DA WEB

847

A segurança da Web pode ser dividida em três partes. Primeiro, como são os objetos e recursos nomeados de forma segura? Em segundo lugar, como pode proteger, conexão autenticada ções sejam estabelecidas? Terceiro, o que acontece quando um site envia uma peça a um cliente de código executável? Depois de examinar algumas ameaças, examinaremos todas essas processa.

8.9.1 Ameacas

Quase todas as semanas, lê-se sobre problemas de segurança do site no jornal. A situação é realmente muito sombria. Vejamos alguns exemplos do que sempre pronto aconteceu. Primeiro, as páginas iniciais de várias organizações estiveram em anexados e substituídos por novas páginas iniciais da escolha dos crackers. (O popular a imprensa chama as pessoas que invadem os computadores de "hackers", mas muitos programadores

reserve esse termo para grandes programadores. Preferimos chamar essas pessoas de "crackers.") Sites que foram invadidos incluem aqueles que pertencem ao Yahoo!, os EUA Exército, CIA, NASA e *New York Times*. Na maioria dos casos, os biscoitos apenas coloquei algum texto engraçado e os sites foram reparados em poucas horas.

Agora, vejamos alguns casos muito mais sérios. Numerosos sites foram derrubado por ataques de negação de serviço, nos quais o cracker inunda o site com o tráfego, tornando-o incapaz de responder a consultas legítimas. Freqüentemente, o ataque é montado a partir de um grande número de máquinas que o cracker já quebrou em (ataques DDoS). Esses ataques são tão comuns que nem mesmo fazem as notícias, mas podem custar aos sites atacados milhares de dólares em perdas o negócio.

Em 1999, um cracker sueco invadiu o site do Hotmail da Microsoft e criou um site espelho que permitia a qualquer pessoa digitar o nome de um usuário do Hotmail e, em seguida, leia todos os e-mails atuais e arquivados da pessoa.

Em outro caso, um cracker russo de 19 anos chamado Maxim invadiu um site de comércio eletrônico e roubou 300.000 números de cartão de crédito. Então ele ap- Procurou os proprietários do site e disse-lhes que, se não lhe pagassem \$ 100.000, ele postaria todos os números de cartão de crédito na Internet. Eles não cederam ao seu chantagem, e ele de fato postou os números do cartão de crédito, causando grandes danos em muitas vítimas inocentes.

Em uma linha diferente, um estudante de 23 anos da Califórnia enviou um comunicado à imprensa para uma agência de notícias afirmando falsamente que a Emulex Corporation postaria um grande perda trimestral e que o CEO estava demitindo-se imediatamente. Dentro de horas, as ações da empresa caíram 60%, fazendo com que os acionistas perdessem mais de \$ 2 bilhões leão. O perpetrador ganhou um quarto de milhão de dólares vendendo as ações a descoberto pouco antes de enviar o anúncio. Embora este evento não fosse um site invasão, é claro que colocar tal anúncio na página inicial de qualquer uma grande corporação teria um efeito semelhante.

Poderíamos (infelizmente) continuar assim por muito mais páginas. Mas é agora hora de examinar alguns dos problemas técnicos relacionados à segurança da Web. Para mais

848
SEGURANÇA DE REDE
INDIVÍDUO. 8

informações sobre problemas de segurança de todos os tipos, consulte Anderson (2008a); Stuttard e Pinto (2007); e Schneier (2004). Pesquisar na Internet também resultará um grande número de casos específicos.

8.9.2 Nomenclatura segura

Vamos começar com algo muito básico: Alice deseja visitar o site de Bob. Ela digita o URL de Bob em seu navegador e, alguns segundos depois, uma página da Web aparece. Mas é do Bob? Talvez sim e talvez não. Trudy pode estar até seu velho truques novamente. Por exemplo, ela pode estar interceptando todos os pacotes de saída de Alice e examiná-los. Quando ela captura uma solicitação HTTP *GET* dirigida a Site de Bob, ela poderia ir ao site de Bob sozinha para obter a página e modificá-la como ela deseja e devolver a página falsa para Alice. Alice não saberia disso. Pior ainda, Trudy poderia cortar os preços na loja virtual de Bob para tornar suas mercadorias muito atraente, enganando assim Alice para enviar seu número de cartão de crédito para "Bob" para comprar alguma mercadoria.

Uma desvantagem desse ataque clássico de intermediário é que Trudy precisa estar em posição de interceptar o tráfego de saída de Alice e forjar seu tráfego de entrada fíco. Na prática, ela precisa grampear a linha telefônica de Alice ou de Bob, já que escutar o backbone de fibra é bastante difícil. Embora a escuta telefônica ativa certamente seja possível, dá muito trabalho e, embora Trudy seja inteligente, ela também é preguiçosa.

Além disso, existem maneiras mais fáceis de enganar Alice.

Spoofing de DNS

Uma maneira seria Trudy quebrar o sistema DNS ou talvez apenas o DNS armazenar em cache no ISP de Alice e substituir o endereço IP de Bob (digamos, 36.1.2.3) pelo dela (Trudy) endereço IP (digamos, 42.9.9.9). Isso leva ao seguinte ataque. O caminho ele deve funcionar é ilustrado na Fig. 8-46 (a). Aqui, Alice (1) pede DNS para O endereço IP de Bob, (2) o obtém, (3) pede a Bob sua página inicial e (4) obtém isso, também. Depois que Trudy modificou o registro DNS de Bob para conter seu próprio endereço IP em vez de Bob, obtemos a situação na Figura 8.46 (b). Aqui, quando Alice olha para cima O endereço IP de Bob, ela obtém o de Trudy, então todo o tráfego destinado a Bob vai para Trudy. Trudy agora pode montar um ataque man-in-the-middle sem ter que ir para o problema de grampear qualquer linha telefônica. Em vez disso, ela precisa invadir um servidor DNS ver e alterar um registro, uma proposição muito mais fácil.

Como Trudy pode enganar o DNS? Acontece que é relativamente fácil. Resumir brevemente marizada, Trudy pode enganar o servidor DNS do ISP de Alice para enviar uma consulta para procurar o endereço de Bob. Infelizmente, como o DNS usa UDP, o servidor DNS não tem uma maneira real de verificar quem forneceu a resposta. Trudy pode explorar isso propriedade forjando a resposta esperada e, assim, injetando um endereço IP falso em o cache do servidor DNS. Para simplificar, vamos supor que o ISP de Alice não inicialmente tem uma entrada para o site de Bob, *bob.com*. Se isso acontecer, Trudy pode esperar até que o tempo limite e tente mais tarde (ou use outros truques).

1. Dê-me o endereço IP de Bob
 2. 36.1.2.3 (endereço IP de Bob)
 3. OBTER index.html
 4. Página inicial de Bob
- Bob's
Rede
servidor
(36.1.2.3)
DNS
servidor
Alice
1
2
3
(uma)
4
1. Dê-me o endereço IP de Bob
 2. 42.9.9.9 (endereço IP de Trudy)
 3. OBTER index.html
 4. A página inicial falsa de Bob de Trudy
- Trudy

Rede
servidor
(42.9.9.9)
Rachado
DNS
servidor
Alice
1
2
3
(b)
4

Figura 8-46. (a) Situação normal. (b) Um ataque baseado na invasão de um DNS servidor e modificar o registro de Bob.

Trudy inicia o ataque enviando uma solicitação de pesquisa ao ISP de Alice solicitando o endereço IP de *bob.com*. Como não há entrada para este nome DNS, o cache O servidor consulta o servidor de nível superior em busca do domínio *com* para obter um. Contudo, Trudy bate o *com* servidor para o soco e envia de volta uma falsa resposta dizendo:

"*bob.com* é 42.9.9.9," onde o endereço IP é dela. Se a falsa resposta dela voltar ao ISP de Alice primeiro, esse será armazenado em cache e a resposta real será rejeitada como uma resposta não solicitada a uma consulta que não está mais pendente. Enganando um servidor DNS para

a instalação de um endereço IP falso é chamada de **falsificação de DNS**. Um cache que contém uma intenção

O endereço IP opcionalmente falso como esse é chamado de **cache envenenado**.

Na verdade, as coisas não são tão simples. Primeiro, o ISP de Alice verifica se a resposta contém o endereço IP de origem correto do servidor de nível superior. Mas desde Trudy pode colocar o que quiser nesse campo de IP, ela pode derrotar esse teste facilmente uma vez que os endereços IP dos servidores de nível superior devem ser públicos.

Em segundo lugar, para permitir que os servidores DNS digam qual resposta vai com qual solicitação, todos

as solicitações carregam um número de sequência. Para enganar o ISP de Alice, Trudy precisa saber seu

número da sequência atual. A maneira mais fácil de aprender o número da sequência atual é para Trudy registrar ela mesma um domínio, digamos, *trudy-the-intruder.com*. Deixe-nos comsume que seu endereço IP também é 42.9.9.9. Ela também cria um servidor DNS para ela domínio *hachurado*, *dns.trudy-the-intruder.com*. Ele também usa o anúncio IP 42.9.9.9 de Trudy vestido, já que Trudy tem apenas um computador. Agora ela tem que fazer o ISP de Alice ciente de seu servidor DNS. Isso é fácil de fazer. Tudo o que ela precisa fazer é perguntar ao ISP de Alice

para *foobar.trudy-the-intruder.com*, o que fará com que o ISP de Alice descubra quem serve novo domínio de Trudy pedindo ao de nível superior *com* servidor.

Página 874

850

SEGURANÇA DE REDE INDIVÍDUO. 8

Com *dns.trudy-the-intruder.com* com segurança no cache do ISP de Alice, o verdadeiro em-a aderência pode começar. Trudy agora consulta o ISP de Alice para obter www.trudy-the-intruder.com.

O ISP naturalmente envia ao servidor DNS de Trudy uma consulta solicitando por isso. Esta consulta carrega o número de sequência que Trudy está procurando. Rápida como um coelho, Trudy pede ao ISP de Alice para procurar Bob. Ela imediatamente responde sua própria pergunta por enviando o ISP uma resposta forjada, supostamente a partir de nível superior *com* servidor, dizendo: "*bob.com* é 42.9.9.9". Esta resposta forjada carrega um número de sequência um acima do que o que ela acabou de receber. Enquanto ela está nisso, ela também pode enviar um segundo para-

gery com um número de sequência dois mais alto, e talvez uma dúzia mais com o aumento números de sequência. Um deles deve combinar. O resto será só

jogado fora. Quando a resposta forjada de Alice chega, ela é armazenada em cache; quando a resposta real

chega mais tarde, é rejeitado porque nenhuma consulta está pendente.

Agora, quando Alice procura `bob.com`, ela é instruída a usar 42.9.9.9, anúncio de Trudy vestir. Trudy montou um ataque man-in-the-middle bem-sucedido da empresa forte de sua própria sala de estar. As várias etapas para este ataque são ilustradas em Fig. 8-47. Este ataque específico pode ser evitado fazendo com que os servidores DNS usem IDs dom em suas consultas, em vez de apenas contar, mas parece que toda vez que um buraco está tapado, aparece outro. Em particular, os IDs têm apenas 16 bits, então trabalhar com todos eles é fácil quando é um computador que está fazendo a suposição ing.

1. Procure `foobar.trudy-the-intruder.com`
(para forçá-lo no cache do ISP)
 2. Procure www.trudy-the-intruder.com
(para obter o próximo número de sequência do ISP)
 3. Solicitar www.trudy-the-intruder.com
(Carregando o próximo número de sequência do ISP, n)
 4. Rápido como um coelho, procure `bob.com`
(para forçar o ISP a consultar o servidor de comunicação na etapa 5)
 5. Consulta legítima para `bob.com` com seq = n + 1
 6. Resposta forjada de Trudy: Bob é 42.9.9.9, seq = n + 1
 7. Resposta real (rejeitada, tarde demais)
- Alice's
ISP's
esconderijo
DNS
servidor
para com
Trudy
5
7
1
2
3
4
6

Figura 8-47. Como Trudy falsifica o ISP de Alice.

DNS seguro

O verdadeiro problema é que o DNS foi projetado em uma época em que a Internet era um centro de pesquisa para algumas centenas de universidades, e nem Alice, nem Bob, nem Trudy foi convidada para a festa. A segurança não era um problema na época; fazendo o Internet work em tudo era o problema. O ambiente mudou radicalmente ao longo do

Página 875

SEC. 8,9
SEGURANÇA DA WEB

851

anos, então, em 1994, a IETF criou um grupo de trabalho para tornar o DNS fundamentalmente seguro

cura. Este projeto (em andamento) é conhecido como **DNSsec** (**segurança DNS**); sua primeira saída

foi apresentado na RFC 2535. Infelizmente, o DNSsec não foi totalmente implantado no entanto, muitos servidores DNS ainda são vulneráveis a ataques de falsificação.

O DNSsec é conceitualmente extremamente simples. É baseado em criptografia de chave pública graficamente. Cada zona DNS (no sentido da Figura 7-5) tem um par de chaves pública / privada. Todas as informações enviadas por um servidor DNS são assinadas com a zona de origem privada chave, para que o destinatário possa verificar a sua autenticidade.

DNSsec oferece três serviços fundamentais:

1. Prova da origem dos dados.
2. Distribuição de chave pública.
3. Transação e autenticação de solicitação.

O serviço principal é o primeiro, que verifica se os dados devolvidos têm sido aprovado pelo proprietário da zona. O segundo é útil para armazenar e recuperar chaves públicas com segurança. O terceiro é necessário para evitar a reprodução e ataques de spoofing. Observe que o sigilo não é um serviço oferecido, pois todos os

as informações no DNS são consideradas públicas. Uma vez que a implementação do DNSsec deve leva vários anos, a capacidade de servidores cientes de segurança para interagir com segurança servidores ty-ignorant são essenciais, o que implica que o protocolo não pode ser alterado. Vejamos agora alguns dos detalhes.

Os registros DNS são agrupados em conjuntos chamados **RRSets** (**R**esource **R**ecord **S**ets), com todos os registros tendo o mesmo nome, classe e tipo sendo agrupados em um conjunto. Um RRSet pode conter vários registros *A*, por exemplo, se um nome DNS resolve para um endereço IP primário e um endereço IP secundário. Os RRSets são ex-tendido com vários novos tipos de registro (discutidos abaixo). Cada RRSet é cripto-hash graficamente (por exemplo, usando SHA-1). O hash é assinado pelo privado da zona chave (por exemplo, usando RSA). A unidade de transmissão para os clientes é o RRSet assinado. Após o recebimento de um RRSet assinado, o cliente pode verificar se ele foi assinado pelo chave privada da zona de origem. Se a assinatura estiver de acordo, os dados serão aceitos. Uma vez que cada RRSet contém sua própria assinatura, os RRSets podem ser armazenados em cache em qualquer lugar, mesmo em servidores não confiáveis, sem colocar a segurança em risco.

DNSsec apresenta vários novos tipos de registro. O primeiro deles é a *CHAVE* registro. Este registro contém a chave pública de uma zona, usuário, host ou outro principal, o algoritmo criptográfico usado para assinatura, o protocolo usado para transmissão, e alguns outros bits. A chave pública é armazenada nua. Certificados X.509 não são usado devido ao seu volume. O campo do algoritmo tem 1 para assinaturas MD5 / RSA (a escolha preferida) e outros valores para outras combinações. O protocolo pode indicar o uso de IPsec ou outros protocolos de segurança, se houver.

O segundo novo tipo de registro é o registro *SIG*. Ele contém o hash assinado de acordo com o algoritmo especificado no registro *KEY*. A assinatura se aplica a todos os registros no RRSet, incluindo quaisquer registros *KEY* presentes, mas excluindo

Página 876

852

SEGURANÇA DE REDE INDIVÍDUO. 8

em si. Também contém os horários em que a assinatura inicia seu período de validade e quando expira, assim como o nome do signatário e alguns outros itens.

O design do DNSsec é tal que a chave privada de uma zona pode ser mantida offline.

Uma ou duas vezes por dia, o conteúdo do banco de dados de uma zona pode ser transferido manualmente

desportivo (por exemplo, em CD-ROM) para uma máquina desconectada na qual a chave privada está

localizado. Todos os RRSets podem ser assinados lá e os registros *SIG* assim produzidos pode ser transportado de volta para o servidor principal da zona em CD-ROM. Desta forma, o a chave privada pode ser armazenada em um CD-ROM trancado em um cofre, exceto quando é inserido-

na máquina desconectada para assinar os novos RRSets do dia. Depois de assinar for concluída, todas as cópias da chave são apagadas da memória e do disco e do O CD-ROM é devolvido ao cofre. Este procedimento reduz a segurança eletrônica para segurança física, algo com que as pessoas entendem como lidar.

Este método de pré-atribuir RRSets acelera muito o processo de resposta consultas, uma vez que nenhuma criptografia precisa ser feita em tempo real. A desvantagem é que um

grande quantidade de espaço em disco é necessária para armazenar todas as chaves e assinaturas no Bancos de dados DNS. Alguns registros aumentarão dez vezes de tamanho devido à assinatura.

Quando um processo do cliente obtém um RRSet assinado, ele deve aplicar a origem chave pública da zona para descriptografar o hash, calcular o próprio hash e comparar o dois valores. Se eles concordarem, os dados são considerados válidos. No entanto, este procedimento levanta a questão de como o cliente obtém a chave pública da zona. Uma maneira é ac-solicite-o de um servidor confiável, usando uma conexão segura (por exemplo, usando IPsec).

No entanto, na prática, espera-se que os clientes sejam pré-configurados com o

chaves públicas de todos os domínios de nível superior. Se Alice agora deseja visitar Bob's Web site, ela pode solicitar o DNS pelo RRSet de *bob.com*, que conterá seu endereço IP e um registro *KEY* contendo a chave pública de Bob. Este RRSet será assinado pelo domínio *com* de nível superior, para que Alice possa verificar facilmente sua validade. Um exemplo do que

este RRSet pode conter é mostrado na Fig. 8-48.

Nome do domínio	
Hora de viver a aula	
Tipo	
Valor	
bob.com.	
86400	
NO	
UMA	
36.1.2.3	
bob.com.	
86400	
NO	
CHAVE	
3682793A7B73F731029CE2737D ...	
bob.com.	
86400	
NO	
SIG	
86947503A8B848F5272E53930C ...	

Figura 8-48. Um exemplo de RRSet para *bob.com*. O registro *KEY* é público de Bob chave. O *SIG* registro é o de nível superior *com* hash de assinatura do servidor do *A* e *KEY* registros para verificar sua autenticidade.

Agora, munida de uma cópia verificada da chave pública de Bob, Alice pode perguntar a Bob's Servidor DNS (executado por Bob) para o endereço IP de www.bob.com. Este RRSet será assinado pela chave privada de Bob, para que Alice possa verificar a assinatura no RRSet Bob retorna. Se Trudy de alguma forma conseguir injetar um RRSet falso em qualquer um dos caches, Alice pode facilmente detectar sua falta de autenticidade porque o registro *SIG* contido nele será incorreto.

Página 877

SEC. 8,9
SEGURANÇA DA WEB

853

No entanto, o DNSsec também fornece um mecanismo criptográfico para vincular um responder a uma consulta específica, para evitar o tipo de spoof que Trudy conseguiu fazer na Fig. 8-47. Esta medida antispoofing (opcional) adiciona à resposta um hash de a mensagem de consulta assinada com a chave privada do respondente. Já que Trudy faz não sei a chave privada de nível superior *com* servidor, ela não pode forjar uma resposta para uma consulta do ISP de Alice enviada para lá. Ela certamente pode obter sua resposta primeiro, mas será rejeitado devido à sua assinatura inválida na consulta com hash.

DNSsec também oferece suporte a alguns outros tipos de registro. Por exemplo, o *CERT* registro pode ser usado para armazenar (por exemplo, X.509) certificados. Este registro foi fornecido porque algumas pessoas querem transformar o DNS em uma PKI. Se isso vai funcionar acontecer, ainda está para ver. Pararemos nossa discussão sobre DNSsec aqui.

Para obter mais detalhes, consulte RFC 2535.

8.9.3 SSL - O Secure Sockets Layer

A nomenclatura segura é um bom começo, mas há muito mais na segurança da web. A próxima etapa é conexões seguras. Agora veremos como as conexões seguras podem ser alcançado. Nada que envolva segurança é simples e isso também não.

Quando a Web explodiu em visualização pública, foi inicialmente usada apenas para distribuir páginas estáticas. No entanto, em pouco tempo, algumas empresas tiveram a ideia de usá-lo para transações financeiras, como compra de mercadorias com cartão de crédito, online bancos e negociação eletrônica de ações. Esses aplicativos criaram uma demanda por conexões seguras. Em 1995, a Netscape Communications Corp., a então dominante fornecedor do navegador, respondeu apresentando um pacote de segurança chamado **SSL (Secure**

Sockets Layer) para atender a essa demanda. Este software e seu protocolo são agora amplamente utilizado, por exemplo, pelo Firefox, Safari e Internet Explorer, por isso vale a pena examinando em alguns detalhes.

SSL constrói uma conexão segura entre dois sockets, incluindo

1. Negociação de parâmetros entre cliente e servidor.
2. Autenticação do servidor pelo cliente.
3. Comunicação secreta.
4. Proteção de integridade de dados.

Já vimos esses itens antes, portanto não há necessidade de elaborá-los.

O posicionamento de SSL na pilha de protocolo usual é ilustrado na Figura 8-49.

Efetivamente, é uma nova camada interposta entre a camada de aplicação e o camada de transporte, aceitando solicitações do navegador e enviando-as para TCP para transmissão ao servidor. Assim que a conexão segura for estabelecida Concluída, a principal tarefa do SSL é lidar com compactação e criptografia. Quando o HTTP é usado sobre SSL, é chamado **HTTPS** (**HTTP seguro**), embora seja o padrão Protocolo HTTP. Às vezes, está disponível em uma nova porta (443) em vez da porta 80.

Página 878

854

SEGURANÇA DE REDE

INDIVÍDUO. 8

Como um aparte, SSL não é restrito a navegadores da Web, mas é o seu mais comum inscrição. Ele também pode fornecer autenticação mútua.

Aplicativo (HTTP)

Segurança (SSL)

Transporte (TCP)

Rede (IP)

Link de dados (PPP)

Físico (modem, ADSL, TV a cabo)

Figura 8-49. Camadas (e protocolos) para um usuário doméstico navegando com SSL.

O protocolo SSL passou por várias versões. Abaixo iremos discutir apenas a versão 3, que é a versão mais usada. SSL suporta uma variedade de opções diferentes. Essas opções incluem a presença ou ausência de compressão, os algoritmos criptográficos a serem usados, e alguns assuntos relacionados à exportação de recursos trições sobre criptografia. O último destina-se principalmente a garantir que criptografia é usada apenas quando ambas as extremidades da conexão estão nos Estados Unidos Estados. Em outros casos, as chaves são limitadas a 40 bits, que os criptógrafos consideram algo como uma piada. A Netscape foi forçada a colocar esta restrição para obter uma licença de exportação do governo dos EUA.

SSL consiste em dois subprotocolos, um para estabelecer uma conexão segura e um para usá-lo. Vamos começar vendo como as conexões seguras são estabelecidas lished. O subprotocolo de estabelecimento de conexão é mostrado na Fig. 8-50. Começa com a mensagem 1 quando Alice envia uma solicitação a Bob para estabelecer uma conexão. A solicitação especifica a versão SSL que Alice tem e suas preferências com respeito para compressão e algoritmos criptográficos. Ele também contém um nonce, R_A , para ser usado mais tarde.

Agora é a vez de Bob. Na mensagem 2, Bob faz uma escolha entre as várias algoritmos que Alice pode apoiar e envia seu próprio nonce, R_B . Então, em mensagem 3, ele envia um certificado contendo sua chave pública. Se este certificado não estiver assinado por alguma autoridade conhecida, ele também envia uma cadeia de certificados que podem ser seguido de volta para um. Todos os navegadores, incluindo o de Alice, vêm pré-carregados com cerca de 100 chaves públicas, então, se Bob puder estabelecer uma cadeia ancorada em uma delas, Alice poderá verificar a chave pública de Bob. Neste ponto, Bob pode enviar algumas outras mensagens (como um pedido de certificado de chave pública de Alice). Quando Bob terminar, ele envia a mensagem 4 para dizer a Alice que é a vez dela.

Alice responde escolhendo uma **chave pré - mestre** aleatória de 384 bits e enviando-a para Bob criptografado com sua chave pública (mensagem 5). A chave de sessão real usada para criptografar dados é derivado da chave pré-mestre combinada com ambos os nonces

de uma forma complexa. Após o recebimento da mensagem 5, Alice e Bob são capaz de calcular a chave de sessão. Por este motivo, Alice diz a Bob para mudar para o

Página 879

SEC. 8,9
SEGURANÇA DA WEB

855

Versão SSL, preferências, R A

Versão SSL, escolhas, R B

Cadeia de certificados X.509

Servidor concluído

E_B (chave pré-mestre)

Mudar cifra

Acabado

Mudar cifra

Acabado

9

7

8

Alice

Prumo

6

5

4

3

2

1

Figura 8-50. Uma versão simplificada do subprotocolo de estabelecimento de conexão SSL.

nova cifra (mensagem 6) e também que ela encerrou o estabelecimento subprotocol (mensagem 7). Bob, então, a reconhece (mensagens 8 e 9).

No entanto, embora Alice saiba quem é Bob, Bob não sabe quem é Alice (a menos que Alice tenha uma chave pública e um certificado correspondente para ela, um improvável

situação para um indivíduo). Portanto, a primeira mensagem de Bob pode muito bem ser um pedido para Alice fazer login usando um nome de login e senha previamente estabelecidos. O protocolo de login, no entanto, está fora do escopo do SSL. Uma vez que foi cumprido realizado, por qualquer meio, o transporte de dados pode começar.

Conforme mencionado acima, o SSL oferece suporte a vários algoritmos criptográficos. O mais forte usa DES triplo com três chaves separadas para criptografia e SHA-1 para integridade da mensagem. Esta combinação é relativamente lenta, por isso é usada principalmente para

aplicações bancárias e outras nas quais a mais alta segurança é exigida. Para outros aplicativos dinâmicos de comércio eletrônico, RC4 é usado com uma chave de 128 bits para criptografia

e MD5 é usado para autenticação de mensagem. RC4 leva a chave de 128 bits como uma semente e o expande para um número muito maior para uso interno. Então ele usa este interno número para gerar um fluxo de chaves. O keystream é XORed com o texto simples para fornecerem uma cifra de fluxo clássica, como vimos na Figura 8-14. As versões de exportação também usam RC4 com chaves de 128 bits, mas 88 dos bits são tornados públicos para tornar o cifra fácil de quebrar.

Para o transporte real, um segundo subprotocolo é usado, conforme mostrado na Fig. 8-51.

As mensagens do navegador são primeiro divididas em unidades de até 16 KB. Se dados

Página 880

856

SEGURANÇA DE REDE

INDIVÍDUO. 8

a compressão é habilitada, cada unidade é então compactada separadamente. Depois disso, um chave secreta derivada de dois nonces e a chave pré-mestre é concatenada com o texto compactado e o resultado é hash com o algoritmo de hash acordado (geralmente MD5). Esse hash é anexado a cada fragmento como o MAC. o fragmento compactado mais MAC é então criptografado com o simétrico acordado algoritmo de criptografia (geralmente por XORing com o keystream RC4). Finalmente, um

o cabeçalho do fragmento é anexado e o fragmento é transmitido pelo TCP connection.

mensagem
autenticação
código
Cabeçalho adicionado
Encriptação
MAC adicionado
Compressão
Fragmentação
Parte 1
Parte 2
Mensagem do navegador

Figura 8-51. Transmissão de dados usando SSL.

Uma palavra de cautela é necessária, no entanto. Uma vez que foi demonstrado que RC4 algumas chaves fracas que podem ser facilmente criptoanalisisadas, a segurança do SSL usando RC4 está em terreno instável (Fluhrer et al., 2001). Navegadores que permitem ao usuário escolher o pacote de criptografia deve ser configurado para usar DES triplo com chaves de 168 bits e SHA-1 o tempo todo, embora essa combinação seja mais lenta do que RC4 e MD5.

Ou, melhor ainda, os usuários devem atualizar para navegadores que suportam o sucessor do SSL que descreveremos em breve.

Um problema com SSL é que os principais podem não ter certificados e até mesmo se o fizerem, nem sempre verificam se as chaves usadas correspondem a eles.

Em 1996, a Netscape Communications Corp. entregou o SSL à IETF para standardização. O resultado foi **TLS (Transport Layer Security)**. É descrito em RFC 5246.

TLS foi construído em SSL versão 3. As mudanças feitas em SSL foram relativamente pequeno, mas apenas o suficiente para que o SSL versão 3 e o TLS não possam interoperar. Para exemplo, a forma como a chave de sessão é derivada da chave pré-mestre e nonces foi

Página 881

SEC. 8,9

SEGURANÇA DA WEB

857

alterado para tornar a chave mais forte (ou seja, mais difícil de criptanalisisar). Por causa disso incompatibilidade, a maioria dos navegadores implementa ambos os protocolos, com o TLS caindo para SSL durante a negociação, se necessário. Isso é conhecido como SSL / TLS. O primeiro

A implementação do TLS apareceu em 1999 com a versão 1.2 definida em agosto de 2008. Inclui suporte para conjuntos de criptografia mais fortes (principalmente AES). SSL permaneceu forte no mercado, embora o TLS provavelmente o substitua gradualmente.

8.9.4 Segurança de código móvel

Nomenclatura e conexões são duas áreas de preocupação relacionadas à segurança da web. Mas existem mais. Nos primeiros dias, quando as páginas da Web eram apenas HTML estático arquivos, eles não continham código executável. Agora, eles geralmente contêm pequenos programas, incluindo miniaplicativos Java, controles ActiveX e JavaScripts. Baixando e executar esse **código móvel** é obviamente um risco de segurança enorme, então vários métodos foram concebidos para minimizá-lo. Vamos agora dar uma olhada rápida em alguns dos problemas levantados pelo código móvel e algumas abordagens para lidar com isso.

Segurança de Applet Java

Applets Java são pequenos programas Java compilados para uma máquina orientada para pilha linguagem chamada **JVM (Java Virtual Machine)**. Eles podem ser colocados em uma web página para download junto com a página. Depois que a página é carregada, os miniaplicativos são inseridos em um interpretador JVM dentro do navegador, conforme ilustrado na Figura 8-52.

Miniaplicativo não confiável
Miniaplicativo confiável
Navegador da web
Caixa de areia
Intérprete
Espaço de endereço virtual
0xFFFFFFFF
0

Figura 8-52. Os miniaplicativos podem ser interpretados por um navegador da web.

A vantagem de executar código interpretado em vez de código compilado é que cada a instrução é examinada pelo intérprete antes de ser executada. Isso dá a

intérprete a oportunidade de verificar se o endereço da instrução é válido. No Além disso, as chamadas do sistema também são capturadas e interpretadas. Como essas chamadas são manuais led é uma questão de política de segurança. Por exemplo, se um miniaplicativo é confiável (por exemplo,

858

SEGURANÇA DE REDE INDIVÍDUO. 8

veio do disco local), suas chamadas de sistema poderiam ser realizadas sem questionamento. No entanto, se um miniaplicativo não for confiável (por exemplo, veio da Internet), pode ser encapsulado no que é chamado de **sandbox** para restringir seu comportamento e prender seu at-tenta usar recursos do sistema.

Quando um miniaplicativo tenta usar um recurso do sistema, sua chamada é passada para um segurança

monitorar para aprovação. O monitor examina a chamada à luz da segurança local política e, em seguida, toma a decisão de permiti-la ou rejeitá-la. Desta forma, é possível dar aos miniaplicativos acesso a alguns recursos, mas não a todos. Infelizmente, a realidade é que o modelo de segurança funciona mal e os bugs aparecem o tempo todo.

ActiveX

Os controles ActiveX são programas binários x86 que podem ser incorporados na Web Páginas. Quando um deles é encontrado, uma verificação é feita para ver se deveria ser executado, e se passar no teste, é executado. Não é interpretado ou colocado em sandbox de qualquer forma, por isso tem tanto poder quanto qualquer outro programa de usuário e pode potencialmente

causar um grande dano. Assim, toda a segurança está na decisão de executar o ActiveX ao controle. Em retrospecto, toda a ideia é uma falha de segurança gigantesca.

O método que a Microsoft escolheu para tomar essa decisão é baseado na ideia de **assinatura de código**. Cada controle ActiveX é acompanhado por uma assinatura digital - um hash do código que é assinado por seu criador usando criptografia de chave pública.

Quando um controle ActiveX aparece, o navegador primeiro verifica a assinatura para certifique-se de que não foi adulterado durante o transporte. Se a assinatura estiver correta, o navegador, em seguida, verifica suas tabelas internas para ver se o criador do programa é confiável ou

há uma cadeia de confiança de volta para um criador confiável. Se o criador for confiável, o pro-grama é executado; caso contrário, não é. O sistema Microsoft para verificar Ac-

Os controles de tiveX são chamados de **Authenticode**.

É útil comparar as abordagens Java e ActiveX. Com o aplicativo Java proach, nenhuma tentativa é feita para determinar quem escreveu o miniaplicativo. Em vez disso, um tempo de execução

intérprete certifica-se de que não faz coisas que o dono da máquina disse applets pode não fazer. Em contraste, com a assinatura de código, não há tentativa de monitorar o comportamento em tempo de execução do código móvel. Se veio de uma fonte confiável e não modificado em trânsito, ele apenas funciona. Nenhuma tentativa é feita para ver se o código é malicioso ou não. Se o programador original *pretendia que* o código formatasse o disco rígido e, em seguida, apague o flash ROM para que o computador nunca mais seja inicializado, e se o programador foi certificado como confiável, o código será executado e destruir o computador (a menos que os controles ActiveX tenham sido desabilitados no navegador).

Muitas pessoas acham que confiar em uma empresa de software desconhecida é assustador. Para demonstrar o problema, um programador em Seattle formou uma empresa de software e o certificado como confiável, o que é fácil de fazer. Ele então escreveu um ActiveX controle que fez um desligamento limpo da máquina e distribuiu seu controle ActiveX trol amplamente. Ele desligou muitas máquinas, mas elas só podiam ser reiniciadas, então não

SEC. 8,9
SEGURANÇA DA WEB

859

mal foi feito. Ele estava apenas tentando expor o problema ao mundo. O de-a resposta oficial foi revogar o certificado para este controle ActiveX específico, que encerrou um curto episódio de constrangimento agudo, mas o problema subjacente ainda está lá para um programador malvado explorar (Garfinkel com Spafford, 2002). Uma vez que não há como policiar as milhares de empresas de software que podem escrever código móvel, a técnica de assinatura de código é um desastre esperando para acontecer.

JavaScript

JavaScript não tem nenhum modelo de segurança formal, mas tem um longo histórico de implementações com vazamento. Cada fornecedor lida com a segurança de uma maneira diferente

maneira. Por exemplo, o Netscape Navigator versão 2 usava algo semelhante ao Java modelo, mas pela versão 4 que havia sido abandonada por um modelo de assinatura de código. O problema fundamental é que permitir que um código estrangeiro seja executado em sua máquina é pedindo problemas. Do ponto de vista da segurança, é como convidar um ladrão para sua casa e, em seguida, tentar observá-lo cuidadosamente para que ele não possa escapar do cozinha na sala de estar. Se algo inesperado acontecer e você estiver dis-tratado por um momento, coisas ruins podem acontecer. A tensão aqui é que o código móvel permite gráficos chamativos e interação rápida, e muitos designers de sites pensam que isso é muito mais importante do que segurança, especialmente quando é alguém outra máquina em risco.

Extensões de navegador

Além de estender as páginas da Web com código, há um mercado em expansão no **extensões , add-ons e plug-ins do navegador** . Eles são programas de computador que estender a funcionalidade dos navegadores da web. Plug-ins geralmente fornecem a capacidade de interpretar ou exibir um determinado tipo de conteúdo, como PDFs ou animações em Flash. Extensões e complementos fornecem novos recursos do navegador, como uma senha melhor gestão, ou formas de interagir com as páginas, por exemplo, marcando-as ou permitindo fácil compra de itens relacionados.

Instalar uma extensão, complemento ou plug-in é tão simples quanto encontrar algo que você deseja ao navegar e seguir o link para instalar o programa.

Esta ação fará com que o código seja baixado da Internet e instalado em o navegador. Todos esses programas são escritos em estruturas que variam de acordo com no navegador que está sendo aprimorado. No entanto, para uma primeira aproximação, eles tornar-se parte da base de computação confiável do navegador. Ou seja, se o código que está instalado tem bugs, o navegador inteiro pode ser comprometido.

Existem dois outros modos de falha óbvios também. O primeiro é que o pro grama pode se comportar de forma maliciosa, por exemplo, ao coletar informações pessoais e enviá-lo para um servidor remoto. Pelo que o navegador sabe, o usuário instalou o extensão precisamente para este propósito. O segundo problema é que os plug-ins fornecem o navegador a capacidade de interpretar novos tipos de conteúdo. Muitas vezes, este conteúdo é um

860
SEGURANÇA DE REDE
INDIVÍDUO. 8

própria linguagem de programação explodida. PDF e Flash são bons exemplos. Quando os usuários visualizam páginas com conteúdo em PDF e Flash, os plug-ins em seus navegadores são ex-ecutando o código PDF e Flash. É melhor que esse código esteja seguro; freqüentemente há vul-capacidades que ele pode explorar. Por todos esses motivos, complementos e plug-ins só deve ser instalado conforme necessário e somente de fornecedores confiáveis.

Vírus

Os vírus são outra forma de código móvel. Apenas, ao contrário dos exemplos acima,

vírus não são convidados de forma alguma. A diferença entre um vírus e um comum O código móvel é que os vírus são escritos para se reproduzirem. Quando um vírus está rives, seja por meio de uma página da Web, um anexo de e-mail ou de alguma outra forma, geralmente começa infectando programas executáveis no disco. Quando um desses gramas é executado, o controle é transferido para o vírus, que geralmente tenta se espalhar para outras máquinas, por exemplo, enviando cópias de si mesmo para todos no catálogo de endereços de e-mail da vítima. Alguns vírus infectam o setor de inicialização do disco rígido, então, quando a máquina é inicializada, o vírus começa a ser executado. Os vírus se tornaram uma grande problema na Internet e causou danos no valor de bilhões de dólares. Não há solução óbvia. Talvez toda uma nova geração de sistemas operacionais sistemas baseados em microkernels seguros e compartimentação rígida de usuários, processos e recursos podem ajudar.

8.10 QUESTÕES SOCIAIS

A Internet e sua tecnologia de segurança é uma área onde as questões sociais, públicas política e tecnologia se enfrentam, muitas vezes com consequências enormes. Abaixo nós examinará brevemente três áreas: privacidade, liberdade de expressão e direitos autorais. Não é preciso dizer que só podemos arranhar a superfície. Para leitura adicional, veja Anderson (2008a), Garfinkel com Spafford (2002) e Schneier (2004). The In-a ternet também está repleta de materiais. Basta digitar palavras como "privacidade" "censura," e "copyright" em qualquer mecanismo de pesquisa. Além disso, consulte o site deste livro para alguns links. Está em <http://www.pearsonhighered.com/tanenbaum>.

8.10.1 Privacidade

As pessoas têm direito à privacidade? Boa pergunta. A Quarta Emenda à Constituição dos EUA proíbe o governo de vasculhar as casas das pessoas, documentos e efeitos sem um bom motivo, e continua a restringir as circunstâncias ao abrigo do qual serão emitidos mandados de busca e apreensão. Assim, a privacidade está no público

agenda por mais de 200 anos, pelo menos nos EUA

O que mudou na última década foi a facilidade com que os governos podem espionar seus cidadãos e a facilidade com que os cidadãos podem prevenir tais

Página 885

SEC. 8,10
PROBLEMAS SOCIAIS

861

espionagem. No século 18, para o governo pesquisar os documentos de um cidadão, ele tinha enviar um policial a cavalo para ir à fazenda do cidadão exigindo ver certos documentos. Foi um procedimento complicado. Hoje em dia, telefone com empresas e provedores de Internet prontamente fornecem gramos quando apresentados a pesquisas garantias. Facilita muito a vida do policial e não há perigo de cair de um cavalo.

A criptografia muda tudo isso. Qualquer um que se dê ao trabalho de descer carregar e instalar o PGP e quem usa uma chave de força alienígena bem guardada pode ser bastante certo de que ninguém no universo conhecido pode ler seu e-mail, mandado de busca e apreensão

ou nenhum mandado de busca. Os governos entendem bem isso e não gostam disso. Real privacidade significa que é muito mais difícil para eles espionar criminosos de todos os tipos, mas é também é muito mais difícil espionar jornalistas e adversários políticos. Consequentemente, alguns governos restringem ou proíbem o uso ou exportação de criptografia. Na França, por exemplo, antes de 1999, toda criptografia foi proibida, a menos que o governo recebeu as chaves.

A França não estava sozinha. Em abril de 1993, o governo dos Estados Unidos anunciou seu intenção de fazer um criptoprocessador de hardware, o **chip clipper**, o padrão para

toda a comunicação em rede. Foi dito que isso garantiria aos cidadãos vacy. Também mencionou que o chip fornecia ao governo a capacidade de descriptografar todo o tráfego por meio de um esquema chamado **depósito de chave**, que permitiu ao governo

acesso a todas as chaves. No entanto, o governo prometeu apenas espionar quando tinha um mandado de busca e apreensão válido. Desnecessário dizer que um grande furor se seguiu, com privacidade

defensores denunciando todo o plano e policiais elogiando-o.

Por fim, o governo recuou e abandonou a ideia.

Uma grande quantidade de informações sobre privacidade eletrônica está disponível no Site da Electronic Frontier Foundation, www.eff.org.

Remailers anônimos

PGP, SSL e outras tecnologias tornam possível que duas partes estabeleçam comunicação segura e autenticada, livre de vigilância de terceiros e inter ference. No entanto, às vezes a privacidade é melhor servida por *não* ter autenticação, na verdade, tornando a comunicação anônima. O anonimato pode ser desejado para mensagens ponto a ponto, grupos de notícias ou ambos.

Vejamos alguns exemplos. Primeiro, dissidentes políticos que vivem sob a autoridade regimes itarianos muitas vezes desejam se comunicar anonimamente para escapar de serem presos ou

morto. Em segundo lugar, irregularidades em muitas empresas, educacionais, governamentais e outras organizações muitas vezes foram expostas por denunciantes, que frequentemente preferir permanecer anônimo para evitar retribuição. Terceiro, pessoas com impopulares visões sociais, políticas ou religiosas podem querer se comunicar através de e-mail ou newsgroups sem se expor. Quarto, as pessoas podem desejar discutir alcoolismo, doença mental, assédio sexual, abuso infantil ou ser um

Página 886

862

SEGURANÇA DE REDE

INDIVÍDUO. 8

membro de uma minoria perseguida em um newsgroup sem ter que ir a público.

Muitos outros exemplos existem, é claro.

Vamos considerar um exemplo específico. Na década de 1990, alguns críticos de um não-tradicional grupo religioso internacional postou suas opiniões em um grupo de notícias da USENET por meio de um

repostador anônimo. Este servidor permitiu aos usuários criar pseudônimos e enviar e-mail para o servidor, que então os reenvia ou reenvia usando os pseudônimos, para que ninguém pudesse dizer de onde as mensagens realmente vieram. Algumas postagens revelam-

ed o que o grupo religioso alegou serem segredos comerciais e documentos com direitos autorais mentos. O grupo religioso respondeu dizendo às autoridades locais que seu comércio segredos foram revelados e seus direitos autorais infringidos, ambos crimes onde o servidor estava localizado. Seguiu-se um processo judicial e o operador do servidor foi compelidos a entregar as informações de mapeamento que revelaram as verdadeiras identidades de as pessoas que fizeram as postagens. (Aliás, esta não foi a primeira vez que um grupo religioso ficou infeliz quando alguém vazou seus segredos comerciais: William Tyndale foi queimado na fogueira em 1536 por traduzir a Bíblia para o inglês lish).

Um segmento substancial da comunidade da Internet ficou completamente indignado com esta quebra de confidencialidade. A conclusão que todos tiraram é que um remailer anônimo que armazena um mapeamento entre endereços de e-mail reais e pseudônimos (agora chamados de repassador tipo 1) não valem muito. Este caso estimulou várias pessoas projetando repassadores anônimos que poderiam resistir a intimações ataques.

Esses novos repassadores, geralmente chamados de **repassadores cypherpunk**, funcionam da seguinte maneira.

O usuário produz uma mensagem de e-mail completa com cabeçalhos RFC 822 (exceto *De :*, é claro), criptografa-o com a chave pública do repassador e o envia para o remailer. Lá, os cabeçalhos RFC 822 externos são removidos, o conteúdo é decriptografada e a mensagem é reenviada. O repassador não tem contas e mantém nenhum registro, portanto, mesmo que o servidor seja confiscado posteriormente, ele não retém nenhum vestígio de mensagens que passaram por ele.

Muitos usuários que desejam anonimato acorrentam suas solicitações por meio de vários repassadores anônimos, como mostrado na Fig. 8-53. Aqui, Alice quer enviar a Bob um Cartão de Dia dos Namorados realmente anônimo, então ela usa três repassadores.

Ela redige a mensagem, M , e coloca um cabeçalho contendo o anúncio de e-mail de Bob vestir. Em seguida, ela criptografa tudo com a chave pública do repassador 3, E_3 (indicada por hachura horizontal). Para isso, ela adiciona um cabeçalho com repassador 3's endereço de e-mail em texto simples. Esta é a mensagem mostrada entre os repassadores 2 e 3 na figura.

Em seguida, ela criptografa esta mensagem com a chave pública do repassador 2, E_2 (indicada por hachura vertical) e adiciona um cabeçalho de texto simples contendo o e-mail do repassador 2 endereço. Essa mensagem é mostrada entre 1 e 2 na Fig. 8-53. Finalmente, ela encriptografa a mensagem inteira com a chave pública do repassador 1, E_1 , e adiciona um prefixo simples

cabeçalho de texto com o endereço de e-mail do repassador 1. Esta é a mensagem mostrada ao direita de Alice na figura e esta é a mensagem que ela realmente transmite.

Página 887

SEC. 8,10
PROBLEMAS SOCIAIS

863

Alice
Prumo
1
Para 1
2
3
Para 2
Repostador anônimo
Criptografado
com E_1
Criptografado
com E_2
Criptografado
com E_3
Para Bob
A 3
M
Para Bob
M
A 3
Para Bob
M
A 3
Para 2
Para Bob
M

Figura 8-53. Como Alice usa três repassadores para enviar uma mensagem a Bob.

Quando a mensagem chega ao repassador 1, o cabeçalho externo é retirado. O corpo é descriptografado e depois enviado por e-mail para o repassador 2. Etapas semelhantes ocorrem nas outras duas repassadores.

Embora seja extremamente difícil para qualquer pessoa rastrear a mensagem final até Alice, muitos repassadores tomam precauções de segurança adicionais. Por exemplo, eles podem reter mensagens por um tempo aleatório, adicionar ou remover lixo no final de uma mensagem e reordenar as mensagens, tudo para tornar mais difícil para qualquer um saber qual mensagem foi enviada

um remailer corresponde a qual entrada, a fim de impedir a análise de tráfego. Para descrição desse tipo de repassador, ver Mazie's e Kaashoek (1998).

O anonimato não se restringe ao e-mail. Também existem serviços que permitem

navegação anônima na Web usando a mesma forma de caminho em camadas em que um nó conhece apenas o próximo nó da cadeia. Este método é chamado **cebola roteamento** para fazer com que cada nó descasque outra camada da cebola para determinar onde for encaminhe o pacote a seguir. O usuário configura seu navegador para usar o serviço anonymizer vice como um proxy. Tor é um exemplo bem conhecido de tal sistema (Dingledine et al., 2004). Daí em diante, todas as solicitações HTTP passam pela rede anonymizer, que solicita a página e a envia de volta. O site vê um nó de saída da rede anonimizadora como a origem da solicitação, não o usuário. Contanto que o rede anonymizer se abstém de manter um registro, depois do fato ninguém pode determinar quem solicitou qual página.

8.10.2 Liberdade de expressão

Privacidade está relacionada a indivíduos que desejam restringir o que outras pessoas podem ver sobre eles. Uma segunda questão social importante é a liberdade de expressão, e seu oposto, censura, que trata de governos que querem restringir o que os indivíduos podem ler e publicar. Com a Web contendo milhões e milhões de páginas, tornar-se o paraíso de um censor. Dependendo da natureza e ideologia do regime, O material banido pode incluir sites que contenham qualquer um dos seguintes:

Página 888

864

SEGURANÇA DE REDE
INDIVÍDUO. 8

1. Material impróprio para crianças ou adolescentes.
2. Ódio dirigido a vários grupos étnicos, religiosos, sexuais ou outros.
3. Informação sobre democracia e valores democráticos.
4. Relatos de eventos históricos contradizendo a versão do governo.
5. Manuais para abrir fechaduras, construir armas, criptografar mensagens, etc.

A resposta usual é banir os sites "ruins".

Às vezes, os resultados são inesperados. Por exemplo, algumas bibliotecas públicas instalaram filtros da Web em seus computadores para torná-los adequados para crianças, bloquear sites de pornografia. Os filtros vetam sites em suas listas negras, mas também verifique se há palavras sujas nas páginas antes de exibi-las. Em um caso em Loudoun County, Virgínia, o filtro bloqueou a busca de um cliente por informações sobre mama câncer porque o filtro viu a palavra "seio". O patrono da biblioteca processou Loudoun Município. No entanto, em Livermore, Califórnia, um pai processou a biblioteca pública por não instalar um filtro depois que seu filho de 12 anos foi pego vendo pornografia há. O que uma biblioteca pode fazer?

Muitas pessoas não sabem que a World Wide Web é uma rede mundial. isto cobre o mundo inteiro. Nem todos os países concordam sobre o que deve ser permitido na rede. Por exemplo, em novembro de 2000, um tribunal francês ordenou que o Yahoo! nia Corporation, para impedir que usuários franceses vejam leilões de memorabilia no site do Yahoo!, porque a posse de tal material viola a lei francesa.

Yahoo! apelou para um tribunal dos EUA, que ficou do lado dele, mas a questão de quais leis aplicar onde está longe de ser resolvida.

Apenas imagine. O que aconteceria se algum tribunal em Utah instruísse a França a bloquear sites que lidam com vinho, porque eles não obedecem às normas de Utah leis mais rígidas sobre o álcool? Suponha que a China exigisse que todos os sites lidar com a democracia seja proibido por não ser do interesse do Estado. Iraniano as leis sobre religião se aplicam à Suécia mais liberal? A Arábia Saudita pode bloquear a web sites que tratam dos direitos das mulheres? A questão toda é uma verdadeira caixa de Pandora. Um comentário relevante de John Gilmore é: "A rede interpreta a censura como danos e rotas em torno dele." Para uma implementação concreta, considere o **serviço nacional** (Anderson, 1996). Seu objetivo é garantir que as informações publicadas não pode ser despublicado ou reescrito, como era comum na União Soviética durante O reinado de Josef Stalin. Para usar o serviço eternity, o usuário especifica por quanto tempo o material deve ser preservado, paga uma taxa proporcional à sua duração e tamanho, e carrega. Depois disso, ninguém pode removê-lo ou editá-lo, nem mesmo o uploader.

Como esse serviço poderia ser implementado? O modelo mais simples é usar um sistema ponto a ponto no qual os documentos armazenados seriam colocados em dezenas de participantes, cada um dos quais recebe uma fração da taxa e, portanto, um incentivo para juntar-se ao sistema. Os servidores devem estar espalhados por muitas jurisdições legais para resiliência máxima. Listas de 10 servidores selecionados aleatoriamente seriam armazenadas

Página 889

SEC. 8,10

PROBLEMAS SOCIAIS

865

com segurança em vários lugares, de modo que, se alguns fossem comprometidos, outros ainda seriam

existir. Uma autoridade empenhada em destruir o documento nunca poderia ter certeza de que encontrou todas as cópias. O sistema também pode ser auto-reparável no sentido de que se soube-se que algumas cópias foram destruídas, os sites restantes seriam tentativa de encontrar novos repositórios para substituí-los.

O serviço da eternidade foi a primeira proposta para um sistema resistente à censura.

Desde então, outras foram propostas e, em alguns casos, implementadas. Vários novos recursos foram adicionados, como criptografia, anonimato e tolerância a falhas.

Muitas vezes, os arquivos a serem armazenados são divididos em vários fragmentos, com cada fragmento

armazenado em muitos servidores. Alguns desses sistemas são Freenet (Clarke et al., 2002), PASIS (Wylie et al., 2000) e Publius (Waldman et al., 2000). De outros trabalho é relatado por Serjantov (2002).

Cada vez mais, muitos países estão tentando regular a exportação de intangíveis, que muitas vezes incluem sites, software, artigos científicos, e-mail, ajuda por telefone - mesas e muito mais. Até mesmo o Reino Unido, que tem uma tradição de liberdade secular da fala, agora está considerando seriamente leis altamente restritivas, que seriam, por exemplo, definir discussões técnicas entre um professor britânico e seu estrangeiro Ph.D. estudante, ambos localizados na Universidade de Cambridge, conforme exportação regulamentada

precisando de uma licença governamental (Anderson, 2002). Escusado será dizer que muitas pessoas considere tal política ultrajante.

Esteganografia

Em países onde a censura é abundante, os dissidentes costumam tentar usar a tecnologia para evitá-lo. A criptografia permite que mensagens secretas sejam enviadas (embora possivelmente não legalmente), mas se o governo pensa que Alice é uma pessoa má, a mera o fato de ela estar se comunicando com Bob pode colocá-la nesta categoria também, pois governos repressivos entendem o conceito de fechamento transitivo, mesmo que eles têm poucos matemáticos. Repostadores anônimos podem ajudar, mas se eles forem proibida internamente e as mensagens para estrangeiros exigem uma exportação do governo licença, eles não podem ajudar muito. Mas a Web pode.

Pessoas que querem se comunicar secretamente, muitas vezes tentam esconder o fato de que qualquer

comunicação em tudo está ocorrendo. A ciência de esconder mensagens é chamada **esteganografia**, das palavras gregas para "escrita encoberta". Na verdade, a antiga Os gregos usavam eles próprios. Heródoto escreveu sobre um general que raspou a cabeça de um mensageiro, tatuou uma mensagem em seu couro cabeludo e deixou o cabelo crescer de novo antes

mandando-o embora. As técnicas modernas são conceitualmente as mesmas, só que têm um maior largura de banda, menor latência e não requerem os serviços de um barbeiro.

Como um caso em questão, considere a Fig. 8-54 (a). Esta fotografia, tirada por um de os autores (AST) no Quênia, contém três zebras contemplando uma árvore de acácia.

Fig. 8-54 (b) parece ser as mesmas três zebras e árvore de acácia, mas tem uma atração extra adicionada. Ele contém o texto completo e integral de cinco de

866

SEGURANÇA DE REDE
INDIVÍDUO. 8

As peças de Shakespeare embutidas nele: *Hamlet*, *King Lear*, *Macbeth*, *The Merchant de Veneza* e *Júlio César*. Juntas, essas peças totalizam mais de 700 KB de texto.

(uma)

(b)

Figura 8-54. (a) Três zebras e uma árvore. (b) Três zebras, uma árvore e a texto completo de cinco peças de William Shakespeare.

Como funciona esse canal esteganográfico? A cor da imagem original é 1024 × 768 pixels. Cada pixel consiste em três números de 8 bits, um para cada intensidade vermelha, verde e azul desse pixel. A cor do pixel é formada pelo superposição linear das três cores. O método de codificação esteganográfica usa o bit de ordem inferior de cada valor de cor RGB como um canal oculto. Assim, cada pixel tem espaço para 3 bits de informação secreta, 1 no valor vermelho, 1 no valor verde e 1 no valor azul. Com uma imagem deste tamanho, até 1024 × 768 × 3 bits ou 294.912 bytes de informações secretas podem ser armazenados nele. O texto completo das cinco peças e um breve aviso somam 734.891 bytes. Isto o texto foi compactado primeiro para cerca de 274 KB usando um algoritmo de compactação padrão ritmo. A saída compactada foi então criptografada usando IDEA e inserida em os bits de ordem inferior de cada valor de cor. Como pode ser visto (ou na verdade, não pode ser visto), a existência da informação é completamente invisível. É igualmente invisível na versão grande e colorida da foto. O olho não consegue distinguir facilmente Cor de 21 bits da cor de 24 bits.

Visualizar as duas imagens em preto e branco com baixa resolução não funciona justiça ao quanto poderosa é a técnica. Para ter uma ideia melhor de como steganografia-obra phy, preparamos uma demonstração, incluindo o full-color high- imagem de resolução da Fig. 8-54 (b) com as cinco peças embutidas nela. Os demônios tration, incluindo ferramentas para inserir e extrair texto em imagens, podem ser encontradas no site do livro.

Para usar a esteganografia para comunicação não detectada, os dissidentes podem criar um site repleto de imagens politicamente corretas, como fotos do Grande Líder, estrelas locais de esportes, cinema e televisão, etc. Claro, as fotos seria crivado de mensagens esteganográficas. Se as mensagens fossem primeiro

SEC. 8,10
PROBLEMAS SOCIAIS

867

compactado e, em seguida, criptografado, mesmo alguém que suspeitou de sua presença teria imensa dificuldade em distinguir as mensagens do ruído branco.

Claro, as imagens devem ser digitalizações recentes; copiando uma imagem da Internet e mudar alguns dos bits é uma indicação inoperante.

As imagens não são, de forma alguma, o único meio de mensagens esteganográficas. Áudio os arquivos também funcionam bem. As informações ocultas podem ser transportadas em uma chamada de voz sobre IP por

manipulando os atrasos do pacote, distorcendo o áudio, ou mesmo nos campos do cabeçalho de pacotes (Lubacz et al., 2010). Até mesmo o layout e a ordem das tags em um O arquivo HTML pode conter informações.

Embora tenhamos examinado a esteganografia no contexto da liberdade de expressão, tem vários outros usos. Um uso comum é para os proprietários de imagens codificar mensagens secretas neles declarando seus direitos de propriedade. Se essa imagem for roubada e colocado em um site, o legítimo proprietário pode revelar a mensagem esteganográfica sage no tribunal para provar de quem é a imagem. Essa técnica é chamada de **marca d'água**. É discutido em Piva et al. (2002).

Para mais informações sobre esteganografia, consulte Wayner (2008).

8.10.3 Copyright

Privacidade e censura são apenas duas áreas onde a tecnologia encontra a política pública cy. Um terceiro é a lei de direitos autorais. **O copyright** é concedido aos criadores do **IP (Propriedade Intelectual)**, incluindo escritores, poetas, artistas, compositores, músicos, fotógrafos, cineastas, coreógrafos e outros, o direito exclusivo para explorar seu IP por algum período de tempo, normalmente a vida do autor mais 50 anos ou 75 anos no caso de propriedade corporativa. Após o copyright de um trabalho expira, ele passa para o domínio público e qualquer pessoa pode usá-lo ou vendê-lo enquanto desejo. O Projeto Gutenberg (www.promo.net/pg), por exemplo, colocou mil areias de obras de domínio público (por exemplo, de Shakespeare, Twain e Dickens) no Rede. Em 1998, o Congresso dos Estados Unidos estendeu os direitos autorais nos Estados Unidos a mais 20 anos a pedido de Hollywood, que afirmou que sem uma prorrogação ninguém criaria mais nada. Em contraste, as patentes duram apenas 20 anos e as pessoas ainda inventam coisas.

Os direitos autorais vieram à tona quando o Napster, um serviço de troca de música, 50 milhões de membros. Embora o Napster não tenha realmente copiado nenhuma música, o tribunais sustentaram que está mantendo um banco de dados central de quem tinha qual música foi contribuída violação de direito, isto é, estava ajudando outras pessoas a infringir. Enquanto ninguém afirma seriamente que copyright é uma má ideia (embora muitos afirmem que o termo está longe muito tempo, favorecendo grandes corporações em relação ao público), a próxima geração de música o compartilhamento já está levantando questões éticas importantes.

Por exemplo, considere uma rede ponto a ponto em que as pessoas compartilham arquivos (música de domínio público, vídeos caseiros, tratados religiosos que não são segredos comerciais, etc.) e talvez alguns protegidos por direitos autorais. Suponha que todos estejam online o tempo via ADSL ou cabo. Cada máquina tem um índice do que está no disco

Página 892

868

SEGURANÇA DE REDE
INDIVÍDUO. 8

disco, além de uma lista de outros membros. Alguém que procura um item específico pode escolher um membro aleatório e veja se ele tem. Se não, ele pode verificar todos os membros em a lista dessa pessoa e todos os membros de suas listas e assim por diante. Computadores são muito bom nesse tipo de trabalho. Tendo encontrado o item, o solicitante apenas o copia. Se o trabalho estiver protegido por direitos autorais, é provável que o solicitante esteja infringindo (embora para transferências internacionais, a questão de qual lei se aplica é importante porque em alguns países o upload é ilegal, mas o download não é). Mas e quanto ao fornecedor? É crime manter músicas pelas quais você pagou e baixou legalmente seu disco rígido onde outras pessoas podem encontrá-lo? Se você tiver uma cabine destrancada no país e um ladrão de IP entram sorrateiramente carregando um notebook e scanner, faz a varredura um livro com direitos autorais para o disco rígido do notebook, e foge, você é culpado de o crime de não proteger os direitos autorais de outra pessoa?

Mas há mais problemas surgindo na área de direitos autorais. Há um enorme morcego o que está acontecendo agora entre Hollywood e a indústria de computadores. O antigo quer proteção rigorosa de toda propriedade intelectual, mas o último não quer para ser o policial de Hollywood. Em outubro de 1998, o Congresso aprovou o **DMCA (Digital Millennium Copyright Act)**, que torna crime contornar qualquer mecanismo de proteção presente em uma obra protegida por direitos autorais ou para informar aos outros como cumvent it. Legislação semelhante foi promulgada na União Europeia. Enquanto

virtualmente ninguém acha que piratas do Extremo Oriente deveriam ter permissão para duplicar obras protegidas por direitos autorais, muitas pessoas pensam que o DMCA muda completamente o equilíbrio

relação entre o interesse do proprietário dos direitos autorais e o interesse público.

Um caso em questão: em setembro de 2000, um consórcio da indústria musical acusado de construir um sistema inquebrável para vender música online patrocinou um concurso levar as pessoas a tentar quebrar o sistema (o que é precisamente a coisa certa a fazer com qualquer novo sistema de segurança). Uma equipe de pesquisadores de segurança de vários universidades

empates, liderados pelo Prof. Edward Felten de Princeton, aceitou o desafio e quebrou o sistema. Eles então escreveram um artigo sobre suas descobertas e o enviaram a um Conferência de segurança USENIX, onde passou por revisão por pares e foi aceita.

Antes de o artigo ser apresentado, Felten recebeu uma carta da Gravadora Industry Association of America que ameaçou processar os autores sob o DMCA se publicaram o jornal.

A resposta deles foi abrir um processo pedindo a um tribunal federal que decidisse se publicar artigos científicos sobre pesquisa de segurança ainda era legal. Temendo uma definição decisiva judicial contra ela, a indústria retirou a ameaça e o tribunal rejeitou

O terno de Felten. Sem dúvida, a indústria foi motivada pela fraqueza de seu caso:

convidou pessoas para tentar quebrar seu sistema e, em seguida, ameaçou processar alguns dos por aceitar seu próprio desafio. Com a ameaça retirada, o jornal foi

publicado (Craver et al., 2001). Um novo confronto é praticamente certo.

Enquanto isso, música e filmes pirateados alimentaram o crescimento massivo de redes ponto a ponto. Isso não agradou aos detentores de direitos autorais, que usaram o DMCA para agir. Agora existem sistemas automatizados que pesquisam ponto a redes peer e, em seguida, disparar avisos para operadores de rede e usuários que são

Página 893

SEC. 8,10

PROBLEMAS SOCIAIS

869

suspeito de infringir direitos autorais. Nos Estados Unidos, esses avisos são conhecidos como **avisos de remoção de DMCA**. Essa busca é uma corrida armamentista porque é difícil capturar infratores de direitos autorais de forma confiável. Até a sua impressora pode ser confundida com um culpado (Piatek et al., 2008).

Uma questão relacionada é a extensão da **doutrina de uso justo**, que foi estabelecida lidas por decisões judiciais em vários países. Esta doutrina diz que os compradores de uma obra protegida por direitos autorais tem certos direitos limitados de copiar a obra, incluindo o direito de citar partes dele para fins científicos, usá-lo como material de ensino em escolas ou faculdades e, em alguns casos, fazer cópias de backup para uso pessoal em caso a mídia original falhe. Os testes para o que constitui uso justo incluem (1) se o uso é comercial, (2) qual porcentagem do todo está sendo copiada, e (3) o efeito da cópia nas vendas da obra. Desde o DMCA e simileis lar da União Europeia proíbem a violação da proteção contra cópia esquemas, essas leis também proíbem o uso justo legal. Com efeito, o DMCA tira direitos históricos dos usuários para dar aos vendedores de conteúdo mais poder. Um grande show para baixo é inevitável.

Outro desenvolvimento em andamento que supera até mesmo o DMCA em sua mudança do equilíbrio entre proprietários de direitos autorais e usuários é uma **computação confiável** como defendido por órgãos da indústria como o TCG (**Trusted Computing Group**), liderado por empresas como Intel e Microsoft. A ideia é fornecer suporte para cuidadosa monitora o comportamento do usuário de várias maneiras (por exemplo, tocando música pirateada) em um nível

abaixo do sistema operacional, a fim de proibir comportamentos indesejados. Isto é realizado com um pequeno chip, chamado de TPM (**Trusted Platform Module**), que é difícil de mexer. A maioria dos PCs vendidos hoje em dia vem equipado com

um TPM. O sistema permite que software escrito por proprietários de conteúdo manipule PCs de maneiras que os usuários não podem alterar. Isso levanta a questão de quem é confiável computação confiável. Certamente, não é o usuário. Escusado será dizer que a consequência social conseqüências deste esquema são imensas. É bom que a indústria finalmente esteja pagando atenção à segurança, mas é lamentável que o motorista esteja cumprindo a lei de direitos autorais em vez de lidar com vírus, crackers, intrusos e outros problemas de segurança que a maioria das pessoas está preocupada.

Em suma, os legisladores e advogados estarão ocupados equilibrando a economia interesses dos proprietários de direitos autorais com o interesse público nos próximos anos. Ciberespaço

não é diferente do espaço para carnes: constantemente coloca um grupo contra outro, sultando em lutas de poder, litígios e (espero), eventualmente, algum tipo de resolução, pelo menos até que apareça alguma nova tecnologia disruptiva.

8.11 RESUMO

A criptografia é uma ferramenta que pode ser usada para manter as informações confidenciais e para garantir sua integridade e autenticidade. Todos os sistemas criptográficos modernos são com base no princípio de Kerckhoff de ter um algoritmo conhecido publicamente e um segredo

Página 894

870

SEGURANÇA DE REDE

INDIVÍDUO. 8

chave. Muitos algoritmos criptográficos usam transformações complexas envolvendo sub-instituições e permutações para transformar o texto simples em texto cifrado. However, se a criptografia quântica puder ser tornada prática, o uso de blocos únicos pode fornecer criptosistemas verdadeiramente inquebráveis.

Os algoritmos criptográficos podem ser divididos em algoritmos de chave simétrica e algoritmos de chave pública. Algoritmos de chave simétrica fragmentam os bits em uma série de rodadas parametrizadas pela chave para transformar o texto simples em texto cifrado. AES (Rijndael) e DES triplo são os algoritmos de chave simétrica mais populares no momento ent. Esses algoritmos podem ser usados no modo de livro de código eletrônico, bloco de criptografia modo de encadeamento, modo de codificação de fluxo, modo de contador e outros.

Algoritmos de chave pública têm a propriedade de que diferentes chaves são usadas para encriptografia e descriptografia e que a chave de descriptografia não pode ser derivada do Chave de encriptação. Essas propriedades permitem publicar a chave pública. o algoritmo de chave pública principal é RSA, que deriva sua força do fato de que é muito difícil fatorar grandes números.

Documentos legais, comerciais e outros precisam ser assinados. Assim, var- esquemas ious foram concebidos para assinaturas digitais, usando tanto a chave simétrica e algoritmos de chave pública. Normalmente, as mensagens a serem assinadas são hash usando algoritmos como SHA-1, e então os hashes são assinados em vez do original mensagens.

O gerenciamento de chave pública pode ser feito usando certificados, que são documentos que vinculam um principal a uma chave pública. Os certificados são assinados por uma autoridade confiável

ou por alguém (recursivamente) aprovado por uma autoridade confiável. A raiz do cadeia deve ser obtida com antecedência, mas os navegadores geralmente têm muitos certificados de raiz

gelo embutido neles.

Essas ferramentas criptográficas podem ser usadas para proteger o tráfego de rede. Operador IPsec na camada de rede, fluxos de pacotes de criptografia de host para host. Firewalls pode filtrar o tráfego que entra ou sai de uma organização, geralmente com base no protocolo e a porta usada. As redes privadas virtuais podem simular uma rede antiga de linha alugada para fornecer certas propriedades de segurança desejáveis. Finalmente, as redes sem fio precisam boa segurança para que todos não leiam todas as mensagens e protocolos como 802.11i pro vide.

Quando duas partes estabelecem uma sessão, elas precisam se autenticar

e, se necessário, estabeleça uma chave de sessão compartilhada. Vários protocolos de autenticação existem, incluindo alguns que usam um terceiro confiável, Diffie-Hellman, Kerberos e criptografia de chave pública.

A segurança de e-mail pode ser alcançada por uma combinação das técnicas que temos estudado neste capítulo. PGP, por exemplo, compacta mensagens e, em seguida, criptografa elas com uma chave secreta e envia a chave secreta criptografada com o pub do receptor chave lic. Além disso, ele também faz o hash da mensagem e envia o hash assinado para verificar integridade da mensagem.

A segurança da Web também é um tópico importante, começando com a nomenclatura segura. DNSsec

fornecendo uma maneira de evitar spoofing de DNS. A maioria dos sites de comércio eletrônico usa

Página 895

SEC. 8,11
RESUMO

871

SSL / TLS para estabelecer sessões seguras e autenticadas entre o cliente e o servidor.

Várias técnicas são usadas para lidar com o código móvel, especialmente sandbox e assinatura de código.

A Internet levanta muitos problemas em que a tecnologia interage fortemente com políticas públicas. Algumas das áreas incluem privacidade, liberdade de expressão e cópia direito.

PROBLEMAS

1. Quebre a seguinte cifra de substituição monoalfabética. O texto simples, consistindo em cartas apenas, é um trecho de um poema de Lewis Carroll.

mvyy bek mnyx n yvjjyr snijrh invq n muvjvt je n idnv
jurhri n fehfevir pyeir oruvdq ki ndq uri jhrnqvdt ed zb jnvy
Irr uem rntrhyb jur yeoijrhi ndq jur jkhjyri nyy nqlndpr
Jurb nhr mnvjjvt ed jur iuvdtyr mvyy bek pezr ndq wevd jur qndpr
mvyy bek, medj bek, mvyy bek, medj bek, mvyy bek wevd jur qndpr
mvyy bek, medj bek, mvyy bek, medj bek, medj bek wevd jur qndpr

2. Uma cifra afim é uma versão de uma cifra de substituição monoalfabética, em que o letters de um alfabeto de tamanho m são primeiros mapeados para os inteiros no intervalo de 0 a $m - 1$. Subsequentemente, o inteiro que representa cada letra do texto simples é transformado em um inteiro representando a letra do texto cifrado correspondente. A função de criptografia para um único carta é $E(x) = (ax + b) \text{ mod } m$, onde m é o tamanho do alfabeto e a e b são o chave da cifra e são co-prime. Trudy descobre que Bob gerou um texto cifrado usando uma cifra afim. Ela obtém uma cópia do texto cifrado e descobre que a maioria letra frequente do texto cifrado é 'R', e a segunda letra mais frequente do texto cifrado é 'K'. Mostre como Trudy pode quebrar o código e recuperar o texto simples.

3. Quebre a seguinte cifra de transposição colunar. O texto simples é retirado de um pop um grande livro de informática, então "computador" é uma palavra provável. O texto simples consiste em farta de letras (sem espaços). O texto cifrado é dividido em blocos de cinco caracteres para legibilidade.

aauan cvlre rurnn dltme aeepb ytust iceat npmey iicgo gorch srsoc
ntii imiha oofpa gsivt tpsit lbovl otoex

4. Alice usou uma cifra de transposição para criptografar suas mensagens para Bob. Para maior segurança, ela criptografou a chave de cifra de transposição usando uma cifra de substituição e manteve o en-cifra criptografada em seu computador. Trudy conseguiu obter o criptografado chave de cifra de transposição. Trudy pode decifrar as mensagens de Alice para Bob? Porque ou porque não?

5. Encontre um teclado de uso único de 77 bits que gere o texto "Hello World" a partir do texto cifrado da Fig. 8-4.

6. Você é um espião e, convenientemente, tem uma biblioteca com um número infinito de livros em sua disposição. Seu operador também tem essa biblioteca à sua disposição. Você concordou

Página 896

872
SEGURANÇA DE REDE
INDIVÍDUO. 8

para usar o *Senhor dos Anéis* como um bloco de uso único. Explique como você pode usar esses ativos para

gerar um bloco único infinitamente longo.

7. A criptografia quântica requer um canhão de fôtons que pode, sob demanda, disparar um único fôton carregando 1 bit. Neste problema, calcule quantos fôtons um bit carrega em um Link de fibra de 250 Gbps. Suponha que o comprimento de um fôton seja igual ao seu comprimento de onda, que, para os fins deste problema, é de 1 micrôn. A velocidade da luz na fibra é 20 cm / nseg.

8. Se Trudy captura e regenera fôtons quando a criptografia quântica está em uso, ela errará em alguns deles e fará com que erros apareçam no bloco único de Bob. o que fração dos bits de pad de uso único de Bob estará com erro, em média?

9. Um princípio criptográfico fundamental afirma que todas as mensagens devem ter redundância. Mas também sabemos que a redundância ajuda um intruso a saber se uma chave adivinhada está correta. Considere duas formas de redundância. Primeiro, os n bits iniciais do texto simples contêm um padrão conhecido. Em segundo lugar, os n bits finais da mensagem contêm um hash sobre a mensagem sóbrio. Do ponto de vista da segurança, esses dois são equivalentes? Discuta sua resposta.

10. Na Fig. 8-6, as caixas P e S se alternam. Embora este arranjo seja estético agradavelmente, é mais seguro do que primeiro ter todas as caixas-P e depois todas as S-caixas? Discuta sua resposta.

11. Projete um ataque ao DES com base no conhecimento de que o texto simples consiste exclusivamente de letras maiúsculas ASCII, mais espaço, vírgula, ponto, ponto e vírgula, carriage return e line feed. Nada se sabe sobre os bits de paridade do texto simples.

12. No texto, calculamos que uma máquina de quebra de cifras com um milhão de processadores que poderia analisar uma chave em 1 nanosegundo levaria 10^{16} anos para quebrar a versão de 128 bits ção da AES. Vamos calcular quanto tempo levará para que esse tempo caia para 1 ano, ainda há muito tempo, claro. Para atingir esse objetivo, precisamos que os computadores sejam 10^{16} vezes Mais rápido. Se a Lei de Moore (o poder de computação dobra a cada 18 meses) continua a esperar, quantos anos vai demorar antes que um computador paralelo possa obter a cifra quebrando o tempo em um ano?

13. O AES suporta uma chave de 256 bits. Quantas chaves o AES-256 tem? Veja se você pode encontrar algum número na física, química ou astronomia de aproximadamente o mesmo tamanho. Use o Internet para ajudar na busca por grandes números. Tire uma conclusão de sua pesquisa.

14. Suponha que uma mensagem tenha sido criptografada usando DES no modo contador. Um pouco de o texto cifrado no bloco C_i é accidentalmente transformado de 0 em 1 durante a transmissão. Como resultado, quanto texto simples ficará distorcido?

15. Agora, considere o encadeamento de blocos de texto cifrado novamente. Em vez de um único bit 0 sendo transformado em 1 bit, um bit 0 extra é inserido no fluxo de texto cifrado após o bloco C_i . Como resultado, quanto texto simples ficará distorcido?

16. Compare o encadeamento de blocos de cifras com o modo de feedback de cifras em termos do número de operações de criptografia necessárias para transmitir um arquivo grande. Qual é mais eficiente e por quanto?

17. Usando o criptosistema de chave pública RSA, com $a = 1, b = 2 \dots y = 25, z = 26$.

(a) Se $p = 5$ e $q = 13$, liste cinco valores legais para d .

Página 897

INDIVÍDUO. 8 PROBLEMAS

873

(b) Se $p = 5, q = 31$ e $d = 37$, encontre e .

(c) Usando $p = 3, q = 11$ e $d = 9$, encontre e e criptografe " olá ".

18. Alice e Bob usam criptografia de chave pública RSA para se comunicarem entre eles. Trudy descobre que Alice e Bob compartilharam um dos primos usados para determinar o número n de seus pares de chaves públicas. Em outras palavras, Trudy descobriu que $n_a = p_a \times q$ e $n_b = p_b \times q$. Como Trudy pode usar essas informações para quebrar o código de Alice?

19. Considere o uso do modo contador, conforme mostrado na Fig. 8-15, mas com $IV = 0$. O uso de 0 ameaça a segurança da cifra em geral?

20. Na Figura 8-20, vemos como Alice pode enviar a Bob uma mensagem assinada. Se Trudy substituir P , Bob pode detectá-lo. Mas o que acontecerá se Trudy substituir P e a assinatura?

21. As assinaturas digitais têm uma fraqueza potencial devido aos usuários preguiçosos. Em e-commerce transações, um contrato pode ser elaborado e o usuário solicitado a assinar seu hash SHA-1.

Se o usuário não verificar de fato se o contrato e o hash correspondem, ele pode inadvertidamente assinar um contrato diferente. Suponha que a Máfia tente explorar isso fraqueza para ganhar algum dinheiro. Eles criaram um site pago (por exemplo, pornografia, jogos bling, etc.) e peça a novos clientes um número de cartão de crédito. Então eles enviam um contrato dizendo que o cliente deseja usar seu serviço e pagar com cartão de crédito e peça ao cliente que assine, sabendo que a maioria deles apenas assinará sem verificar ver se o contrato e o hash estão de acordo. Mostre como a máfia pode comprar diamantes de um

joalheiro legítimo da Internet e cobrá-los de clientes desavisados.

22. Uma classe de matemática tem 25 alunos. Supondo que todos os alunos nasceram no primeiro metade do ano - entre 1º de janeiro e 30 de junho - qual é a probabilidade de que em pelo menos dois alunos fazem aniversário no mesmo dia? Suponha que ninguém nasceu em um dia bissexto, portanto, há 181 aniversários possíveis.

23. Depois que Ellen confessou a Marilyn sobre tê-la enganado na questão da posse de Tom, Marilyn resolveu evitar esse problema ditando o conteúdo de mensagens futuras em uma máquina de ditar e ter sua nova secretária apenas digitando. Marilyn então planejou examinar as mensagens em seu terminal depois que eles foram digitados para fazer certeza de que continham suas palavras exatas. A nova secretária ainda pode usar a data de aniversário em-tática para falsificar uma mensagem e, em caso afirmativo, como? *Dica* : ela pode.

24. Considere a tentativa fracassada de Alice de obter a chave pública de Bob na Figura 8.23. Suponha que Bob e Alice já compartilham uma chave secreta, mas Alice ainda quer a chave pública de Bob. É existe agora uma maneira de obtê-lo com segurança? Se sim, como?

25. Alice deseja se comunicar com Bob, usando criptografia de chave pública. Ela estabelece Ela estabelece uma conexão com alguém que ela espera que seja Bob. Ela pede a ele sua chave pública e ele o envia a ela em texto simples, junto com um certificado X.509 assinado pela CA raiz. Alice já possui a chave pública da CA raiz. Quais etapas Alice realiza para verificar se ela está falando com Bob? Suponha que Bob não se importe com quem está falando (por exemplo, Bob é algum tipo de serviço público).

26. Suponha que um sistema use PKI com base em uma hierarquia estruturada em árvore de CAs. Alice deseja se comunicar com Bob e recebe um certificado de Bob assinado por uma CA X após estabelecer um canal de comunicação com Bob. Suponha que Alice nunca ouviu falar de X . Quais etapas Alice executa para verificar se ela está falando com Bob?

Página 898

874

SEGURANÇA DE REDE INDIVÍDUO. 8

27. O IPsec usando AH pode ser usado no modo de transporte se uma das máquinas estiver atrás de um NAT? caixa? Explique sua resposta.

28. Alice deseja enviar uma mensagem para Bob usando hashes SHA-1. Ela consulta com você quanto ao algoritmo de assinatura apropriado a ser usado. O que você sugeriria?

29. Dê um motivo pelo qual um firewall pode ser configurado para inspecionar o tráfego de entrada. Dar um motivo pelo qual ele pode ser configurado para inspecionar o tráfego de saída. Você acha que o as inspeções têm probabilidade de ser bem-sucedidas?

30. Suponha que uma organização use VPN para conectar com segurança seus sites na Internet. Jim, um usuário da organização usa a VPN para se comunicar com sua chefe, Mary. Descreva um tipo de comunicação entre Jim e Mary que não requer uso de criptografia ou outro mecanismo de segurança, e outro tipo de comunicação que exigiria criptografia ou outros mecanismos de segurança. Explique sua resposta.

31. Altere uma mensagem no protocolo da Fig. 8-34 de forma secundária para torná-la resistente a o ataque de reflexão. Explique por que sua mudança funciona.

32. A troca de chaves Diffie-Hellman está sendo usada para estabelecer uma chave secreta entre Alice e Bob. Alice envia Bob $(227, 5, 82)$. Bob responde com (125) . Alice's o número secreto, x , é 12 e o número secreto de Bob, y , é 3. Mostre como Alice e Bob calcule a chave secreta.

33. Dois usuários podem estabelecer uma chave secreta compartilhada usando o algoritmo Diffie-Hellman, mesmo se eles nunca se conheceram, não compartilham segredos e não têm certificados
(a) Explique como esse algoritmo é suscetível a um ataque man-in-the-middle.
(b) Como essa suscetibilidade mudaria se n ou g fossem secretos?

34. No protocolo da Figura 8-39, por que A é enviado em texto simples junto com o sessão-chave?

35. No protocolo Needham-Schroeder, Alice gera dois desafios, R_A e R_{A2} . Isso parece um exagero. Alguém não teria feito o trabalho?

36. Suponha que uma organização use Kerberos para autenticação. Em termos de segurança e disponibilidade do serviço, qual é o efeito se o AS ou TGS cair?

37. Alice está usando o protocolo de autenticação de chave pública da Figura 8-43 para autenticar com comunicação com Bob. No entanto, quando o envio de mensagem 7, Alice esqueceu de criptografar R_B . Trudy agora sabe o valor de R_B . Alice e Bob precisam repetir a autenticação procedimento de instalação com novos parâmetros para garantir uma comunicação segura? Ex-claramente sua resposta.

38. No protocolo de autenticação de chave pública da Figura 8-43, na mensagem 7, R_B é criptografado com K_S . Essa criptografia é necessária ou seria adequado enviá-la de volta em texto simples? Explique sua resposta.

39. Terminais de ponto de venda que usam cartões de tarja magnética e códigos PIN têm uma falha fatal:

um comerciante malicioso pode modificar seu leitor de cartão para registrar todas as informações no cartão e o código PIN para postar transações adicionais (falsas) no futuro. Próximos terminais de geração usarão cartões com CPU completa, teclado e tela minúscula no cartão. Desenvolva um protocolo para este sistema que comerciantes mal-intencionados não possam quebrar.

Página 899

INDIVÍDUO. 8

PROBLEMAS

875

40. É possível fazer o multicast de uma mensagem PGP? Que restrições seriam aplicadas?
41. Supondo que todos na Internet usassem PGP, uma mensagem PGP poderia ser enviada a um endereço de Internet arbitrário e ser decodificado corretamente por todos os envolvidos? Discuta o seu resposta.
42. O ataque mostrado na Fig. 8-47 deixa de fora uma etapa. A etapa não é necessária para o falsificação para funcionar, mas incluí-lo pode reduzir a suspeita potencial após o fato. O que é a etapa que falta?
43. O protocolo de transporte de dados SSL envolve dois nonces, bem como uma chave pré-mestre. Qual valor, se houver, tem o uso de nonces?
44. Considere uma imagem de 2048×512 pixels. Você deseja criptografar um arquivo de 2,5 MB. Que fração do arquivo você pode criptografar nesta imagem? Que fração você seria capaz de criptografar se você compactou o arquivo para um quarto de seu tamanho original? Mostre seus cálculos.
45. A imagem da Fig. 8-54 (b) contém o texto ASCII de cinco peças de Shakespeare. Seria possível esconder música entre as zebras em vez de texto? Se sim, como seria funcionou e quanto você poderia esconder nesta foto? Se não, porque não?
46. Você recebe um arquivo de texto de 60 MB, que deve ser criptografado usando esteganografia nos bits de ordem inferior de cada cor em um arquivo de imagem. Qual seria o tamanho da imagem necessário para criptografar o arquivo inteiro? Qual tamanho seria necessário se o arquivo fosse primeiro compactado para um terço de seu tamanho original? Dê sua resposta em pixels e mostre seus cálculos. Suponha que as imagens tenham uma proporção de 3: 2, por exemplo, 3000×2000 pixels.
47. Alice era uma usuária intensa de um repostador anônimo tipo 1. Ela postaria muitas mensagens sages ao seu newsgroup favorito, *alt.fanclub.alice*, e todos saberiam de todos eles veio de Alice porque todos eles tinham o mesmo pseudônimo. Supondo que o remailer funcionou corretamente, Trudy não conseguiu se passar por Alice. Depois de repostadores tipo 1 foram todos desligados, Alice mudou para um repostador cypherpunk e iniciou um novo tópico em seu newsgroup. Pense em uma maneira de ela impedir que Trudy poste novas mensagens para o grupo de notícias, personificando Alice.
48. Pesquise na Internet por um caso interessante envolvendo privacidade e escreva uma revisão de uma página porta nele.
49. Pesquise na Internet por algum processo judicial envolvendo direitos autorais versus uso justo e escreva um Relatório de uma página resumindo suas descobertas.
50. Escreva um programa que criptografe sua entrada por XORing com um fluxo de chaves. Encontre ou escreva o melhor gerador de números aleatórios possível para gerar o fluxo de chaves. O programa deve atuar como um filtro, pegando o texto simples na entrada padrão e produzindo o texto cifrado na saída padrão (e vice-versa). O programa deve ter um parâmetro, a chave que semeia o gerador de números aleatórios.
51. Escreva um procedimento que calcule o hash SHA-1 de um bloco de dados. O procedimento deve ter dois parâmetros: um ponteiro para o buffer de entrada e um ponteiro para um 20 bytes buffer de saída. Para ver a especificação exata do SHA-1, pesquise FIPS na Internet 180-1, que é a especificação completa.

Página 900

876

SEGURANÇA DE REDE

INDIVÍDUO. 8

52. Escreva uma função que aceite um fluxo de caracteres ASCII e criptografe esta entrada usando uma cifra de substituição com o modo Cipher Block Chaining. O tamanho do bloco deve ter 8 bytes. O programa deve obter texto simples da entrada padrão e imprima o texto cifrado na saída padrão. Para este problema, você tem permissão para selecionar qualquer sistema razoável para determinar que o fim da entrada é alcançado, e / ou quando preenchimento deve ser aplicado para completar o bloqueio. Você pode selecionar qualquer formato de saída, contanto que seja inequívoco. O programa deve receber dois parâmetros:
1. Um ponteiro para o vetor de inicialização; e

2. Um número, k , que representa o deslocamento da cifra de substituição, de modo que cada caractere ASCII O acter seria criptografado pelo k -ésimo caractere à frente dele no alfabeto.
Por exemplo, se $x = 3$, então A é codificado por D, B é codificado por E etc. Make reasuposições aceitáveis com relação a alcançar o último caractere no conjunto ASCII.
Certifique-se de documentar claramente em seu código quaisquer suposições que você fizer sobre o algoritmo de entrada e criptografia.

53. O objetivo deste problema é dar-lhe uma melhor compreensão quanto ao mecanismo ismos de RSA. Escreva uma função que recebe como seus parâmetros primos p e q , calculates chaves RSA públicas e privadas usando esses parâmetros, e produz n , z , d e e como impressões para a saída padrão. A função também deve aceitar um fluxo de ASCII caracteres e criptografar essa entrada usando as chaves RSA calculadas. O programa deve pegue o texto simples da entrada padrão e imprima o texto cifrado na saída padrão.
A criptografia deve ser realizada em termos de caracteres, ou seja, tomar cada caractere no entrada e criptografá-lo independentemente de outros caracteres na entrada. Para este problema, você tem permissão para selecionar qualquer sistema razoável para determinar que o final da entrada é atingido. Você pode selecionar qualquer formato de saída, desde que não seja ambíguo. Faço certifique-se de documentar claramente em seu código todas as suposições que você fizer sobre a entrada e Algoritmo de criptografia.

Página 901

9

LISTA DE LEITURA E BIBLIOGRAFIA

Agora terminamos nosso estudo de redes de computadores, mas este é apenas o descaroçamento. Muitos tópicos interessantes não foram tratados com tantos detalhes quanto merecem, e outros foram totalmente omitidos por falta de espaço. Neste capítulo depois, fornecemos algumas sugestões para leituras adicionais e uma bibliografia, para o benefício dos leitores que desejam continuar seus estudos de redes de computadores.

9.1 SUGESTÕES PARA LEITURA ADICIONAL

Existe uma vasta literatura sobre todos os aspectos das redes de computadores. Dois periódicos que publicam artigos nesta área são *IEEE / ACM Transactions on Networking* e *IEEE Journal on Selected Areas in Communications*.

Os periódicos dos grupos de interesse especial da ACM em comunicações de dados (SIGCOMM) e Mobilidade de Sistemas, Usuários, Dados e Computação (SIGMOBILE) publicam muitos artigos de interesse, especialmente sobre tópicos emergentes. Eles são *Revisão de comunicação por computador e computação e comunicações móveis Revisão*.

O IEEE também publica três revistas - *IEEE Internet Computing*, *IEEE Network Magazine* e *IEEE Communications Magazine* —que contêm pesquisas, tutoriais e estudos de caso sobre redes. Os dois primeiros enfatizam a arquitetura, padrões e software, e o último tende para a tecnologia de comunicação (fibra óptica, satélites e assim por diante).

877

Página 902

878

LISTA DE LEITURA E BIBLIOGRAFIA
INDIVÍDUO. 9

Há uma série de conferências anuais ou semestrais que atraem numerosos artigos em redes. Em particular, procure a conferência *SIGCOMM*, *NSDI* (Simpósio sobre Projeto e Implementação de Sistemas em Rede), *MobiSys* (Conferência sobre Sistemas Móveis, Aplicativos e Serviços), *SOSP* (Simpósio sobre Princípios de Sistemas Operacionais) e *OSDI* (Simpósio sobre Sistemas Operacionais

Design e implementação).

Abaixo listamos algumas sugestões de leitura suplementar, relacionadas ao capítulo res deste livro. Muitas das sugestões são livros de capítulos em livros, com alguns tutoriais e pesquisas. As referências completas estão na Seç. 9.2.

9.1.1 Introdução e Obras Gerais

Comer, *The Internet Book*, 4^a ed.

Quem procura uma introdução fácil à Internet deve procurar aqui. Comer descreve a história, crescimento, tecnologia, protocolos e serviços de a Internet em termos que os novatos podem entender, mas muito material é coberto que o livro também é do interesse de leitores mais técnicos.

Computer Communication Review, edição do 25º aniversário, janeiro de 1995

Para uma visão em primeira mão de como a Internet se desenvolveu, esta edição especial coleta artigos importantes até 1995. Incluem-se artigos que mostram o desenvolvimento de TCP, multicast, DNS, Ethernet e a arquitetura geral.

Crovella and Krishnamurthy, *Internet Measurement*

Como sabemos se a Internet funciona bem? Esta questão não é trivial de responder porque ninguém está no comando da Internet. Este livro descreve as técnicas que foram desenvolvidas para medir o funcionamento da Internet, da infraestrutura de rede aos aplicativos.

IEEE Internet Computing, janeiro a fevereiro. 2000

A primeira edição do *IEEE Internet Computing* no novo milênio fez exatamente o que você esperaria: perguntou às pessoas que ajudaram a criar a Internet no milênio anterior para especular sobre onde está indo no próximo. Os especialistas são Paul Baran, Lawrence Roberts, Leonard Kleinrock, Stephen Crocker, Danny Cohen, Bob Metcalfe, Bill Gates, Bill Joy e outros. Veja quão bem seus

as previsões aconteceram mais de uma década depois.

Kipnis, " Batendo o Sistema: Abusos do Processo de Adoção de Padrões "

Os comitês de padrões tentam ser justos e neutros em relação ao fornecedor em seu trabalho, mas não

felizmente existem empresas que tentam abusar do sistema. Por exemplo, tem aconteceu repetidamente que uma empresa ajuda a desenvolver um padrão e depois é aprovado, anuncia que o padrão é baseado em uma patente que possui e da qual vai licenciar para empresas de que gosta e não para empresas de que não gosta, em

Página 903

SEC. 9,1

SUGESTÕES PARA LEITURA ADICIONAL

879

preços que ela sozinha determina. Para dar uma olhada no lado negro da padronização, este artigo é um excelente começo.

Hafner e Lyon, *onde os feiticeiros ficam acordados até tarde*

Naughton, *uma breve história do futuro*

Quem inventou a Internet, afinal? Muitas pessoas reivindicaram o crédito. E com razão, visto que muitas pessoas participaram, de maneiras diferentes. Estava Paul Baran, que escreveu um relatório descrevendo a troca de pacotes, havia o pessoal da várias universidades que projetaram a arquitetura ARPANET, havia o pessoas da BBN que programaram os primeiros IMPs, havia Bob Kahn e Vint Cerf que inventou o TCP / IP e assim por diante. Esses livros contam a história da Internet, pelo menos até 2.000, repleto de muitas anedotas.

9.1.2 A Camada Física

Bellamy, *Digital Telephony*, 3rd ed.

Para dar uma olhada naquela outra rede importante, a rede telefônica, este livro oficial contém tudo que você sempre quis saber e muito mais. Particularmente interessantes são os capítulos sobre transmissão e multiplexação, comutação, fibra óptica, telefonia móvel e DSL.

Hu e Li, " Internet baseada em satélite: um tutorial "

O acesso à Internet via satélite é diferente do uso de linhas terrestres. Não somente

existe a questão do atraso, mas o roteamento e a comutação também são diferentes. Nisso Neste artigo, os autores examinam as questões relacionadas ao uso de satélites para acesso à Internet. Joel, " Telecommunications and the IEEE Communications Society "

Para uma história compacta, mas surpreendentemente abrangente das telecomunicações, começando com o telégrafo e terminando com 802.11, este artigo é o lugar para procurar. Também cobre rádio, telefones, comutação analógica e digital, cabos submarinos, transmissão digital, transmissão de televisão, satélites, TV a cabo, comunicação óptica comunicações, telefones celulares, comutação de pacotes, a ARPANET e a Internet.

Palais, *Fiber Optic Communication*, 5^a ed.

Livros sobre tecnologia de fibra óptica costumam ser voltados para o especialista, mas este é mais acessível do que a maioria. Abrange guias de ondas, fontes de luz, detectores de luz, acopladores, modulação, ruído e muitos outros tópicos.

Su, a *interface aérea UMTS em engenharia de RF*

Este livro fornece uma visão geral detalhada de um dos principais sistemas celulares 3G tems. É focado na interface aérea, ou protocolos sem fio que são usados entre celulares e a infraestrutura de rede.

Página 904

880

LISTA DE LEITURA E BIBLIOGRAFIA

INDIVÍDUO. 9

Quer, *RFID explicou*

O livro de Want é uma cartilha de fácil leitura sobre como a tecnologia incomum do A camada física RFID funciona. Abrange todos os aspectos de RFID, incluindo seu potencial formulários. Alguns exemplos do mundo real de implantações de RFID e a experiência ganho com eles também é convertido.

9.1.3 A camada de enlace de dados

Kasim, *entregando Carrier Ethernet*

Hoje em dia, a Ethernet não é apenas uma tecnologia local. A nova moda é use Ethernet como um link de longa distância para Ethernet de nível de operadora. Este livro traz juntos ensaios para cobrir o tópico em profundidade.

Lin e Costello, *Error Control Coding*, 2^a ed.

Os códigos para detectar e corrigir erros são fundamentais para redes de computadores confiáveis. Este livro popular explica alguns dos códigos mais importantes, desde códigos Hamming lineares para códigos de verificação de paridade de baixa densidade mais complexos. Tenta

fazer isso com o mínimo de álgebra necessário, mas ainda é muito.

Stallings, *Data and Computer Communications*, 9^a ed.

A segunda parte cobre a transmissão digital de dados e uma variedade de links, incluindo erros detecção, controle de erros com retransmissões e controle de fluxo.

9.1.4 A subcamada de controle de acesso do meio

Andrews et al., *Fundamentals of WiMAX*

Este livro abrangente oferece um tratamento definitivo da tecnologia WiMAX, da ideia de banda larga sem fio, às técnicas sem fio usando OFDM e multiplas antenas, através do sistema multi-acesso. Seu estilo de tutorial dá sobre o tratamento mais acessível que você encontrará para este material pesado.

Gast, *802.11 Wireless Networks*, 2^a ed.

Para uma introdução legível à tecnologia e protocolos de 802.11, este é um bom lugar para começar. Ele começa com a subcamada MAC e, em seguida, introduz o material na as diferentes camadas físicas e também de segurança. No entanto, a segunda edição não é novo o suficiente para ter muito a dizer sobre 802.11n.

Perlman, *Interconexões*, 2^a ed.

Para um tratamento confiável, mas divertido, de pontes, roteadores e roteamento em geral, o livro de Perlman é o lugar para procurar. O autor projetou o algoritmos usados no IEEE 802 spanning tree bridge e ela é uma das principais autoridades em vários aspectos da rede.

SEC. 9,1
SUGESTÕES PARA LEITURA ADICIONAL

881

9.1.5 A Camada de Rede

Comer, *Internetworking with TCP / IP*, Vol. 1, 5^a ed.

Comer escreveu o trabalho definitivo sobre o pacote de protocolos TCP / IP, agora em seu Quinta edição. A maior parte da primeira metade lida com IP e protocolos relacionados na rede camada de trabalho. Os outros capítulos tratam principalmente das camadas superiores e também são Vale a pena ler.

Grayson et al., *IP Design for Mobile Networks*

As redes telefônicas tradicionais e a Internet estão em rota de colisão, com redes de telefonia móvel sendo implementadas com IP interno. Este livro conta como projetar uma rede usando os protocolos IP que suportam telefone móvel serviço.

Huitema, *Routing in the Internet*, 2^a ed.

Se você deseja obter uma compreensão profunda dos protocolos de roteamento, este é um bom livro. Ambos os algoritmos pronunciáveis (por exemplo, RIP e CIDR) e não reproduzidos algoritmos substantiváveis (por exemplo, OSPF, IGRP e BGP) são tratados em detalhes.

Desenvolvimentos mais recentes não são cobertos uma vez que este é um livro mais antigo, mas o que é

coberto é explicado muito bem.

Koodli e Perkins, *interconexão móvel com IPv6*

Dois desenvolvimentos importantes da camada de rede são apresentados em um volume: IPv6 e IP móvel. Ambos os tópicos são bem cobertos, e Perkins foi um dos forças motrizes por trás do IP móvel.

Nucci e Papagiannaki, *Projeto, Medição e Gerenciamento de Grande Escala Redes IP*

Conversamos muito sobre como as redes funcionam, mas não como você assine, implante e gerencie um se você fosse um ISP. Este livro preenche essa lacuna, procurando em métodos modernos para engenharia de tráfego e como os ISPs fornecem serviços usando redes.

Perlman, *Interconexões*, 2^a ed.

No Chaps. 12 a 15, Perlman descreve muitos dos problemas envolvidos na projeto de algoritmo de roteamento unicast e multicast, tanto para redes de longa distância quanto redes de LANs. Mas, de longe, a melhor parte do livro é o cap. 18, em que o autora destila seus muitos anos de experiência com protocolos de rede em uma capítulo mativo e divertido. É leitura obrigatória para designers de protocolo.

Stevens, *TCP / IP Illustrated*, Vol. 1

Os capítulos 3–10 fornecem um tratamento abrangente de IP e protocolos relacionados (ARP, RARP e ICMP), ilustrados por exemplos.

882

LISTA DE LEITURA E BIBLIOGRAFIA
INDIVÍDUO. 9

Varghese, *algoritmos de rede*

Passamos muito tempo falando sobre como roteadores e outros elementos de rede mentos interagem uns com os outros. Este livro é diferente: é sobre como os roteadores são realmente projetado para encaminhar pacotes em velocidades prodigiosas. Para informações privilegiadas sobre

isso e outras questões relacionadas, este é o livro para ler. O autor é uma autoridade em algoritmos inteligentes que são usados na prática para implementar elementos de rede de alta velocidade

mentos em software e hardware.

9.1.6 A Camada de Transporte

Comer, *Internetworking with TCP / IP*, Vol. 1, 5^a ed.

Como mencionado acima, Comer escreveu o trabalho definitivo sobre o TCP / IP conjunto de protocolos. A segunda metade do livro é sobre UDP e TCP.

Farrell e Cahill, *Rede Tolerante a Delay e Disruption*

Este pequeno livro é o único a ser lido para uma visão mais profunda da arquitetura, proto cols, e aplicações de "redes desafiadas" que devem operar sob condições condições de conectividade. Os autores participaram do desenvolvimento de DTNs no IETF DTN Research Group.

Stevens, *TCP / IP Illustrated*, Vol. 1

Os capítulos 17-24 fornecem um tratamento abrangente do TCP ilustrado por exemplos.

9.1.7 A Camada de Aplicação

Berners-Lee et al., "The World Wide Web"

Faça uma viagem no tempo para ter uma perspectiva sobre a Web e para onde ela está indo a pessoa que o inventou e alguns de seus colegas do CERN. O artigo concentra-se na arquitetura da Web, URLs, HTTP e HTML, bem como di- e compara com outros sistemas de informação distribuídos.

Held, *A Practical Guide to Content Delivery Networks*, 2nd ed.

Este livro oferece uma exposição prática de como funcionam os CDNs, enfatizando as considerações práticas ao projetar e operar um CDN com bom desempenho.

Hunter et al., *Beginning XML*, 4^a ed.

Existem muitos, muitos livros sobre HTML, XML e serviços da Web. Este 1000- O livro de páginas cobre a maior parte do que você provavelmente deseja saber. Não explica apenas como escrever XML e XHTML, mas também como desenvolver serviços da Web que produzir e manipular XML usando Ajax, SOAP e outras técnicas que são comumente usado na prática.

Página 907

SEC. 9,1

SUGESTÕES PARA LEITURA ADICIONAL

883

Krishnamurthy e Rexford, *Web Protocols and Practice*

Seria difícil encontrar um livro mais abrangente sobre todos os aspectos do Web do que este. Ele cobre clientes, servidores, proxies e cache, como você pode Espero. Mas também há capítulos sobre o tráfego da Web e medições, bem como capítulos sobre pesquisas atuais e melhorias na Web.

Simpson, *Video Over IP*, 2^a ed.

O autor analisa amplamente como a tecnologia IP pode ser usada para mover vídeo em redes, tanto na Internet quanto em redes privadas destinadas a carregue o vídeo. Curiosamente, este livro é voltado para o aprendizado profissional de vídeo sobre networking, e não o contrário.

Wittenburg, *Comprendendo a tecnologia de voz sobre IP*

Este livro cobre como funciona a voz sobre IP, desde o transporte de dados de áudio com o Protocolos IP e questões de qualidade de serviço, até o conjunto SIP e H.323 de protocolos. É necessariamente detalhado dado o material, mas acessível e quebrado em unidades digestíveis.

9.1.8 Segurança de rede

Anderson, *Engenharia de Segurança*, 2º. ed.

Este livro apresenta uma maravilhosa combinação de técnicas de segurança expressas em uma compreensão de como as pessoas os usam (e abusam). É mais técnico do que *Segredos e mentiras*, mas menos técnicos do que *Segurança de rede* (veja abaixo). Depois de um introdução às técnicas básicas de segurança, capítulos inteiros são dedicados a vários ous aplicativos, incluindo bancos, comando e controle nuclear, impressão de segurança ing, biometria, segurança física, guerra eletrônica, segurança de telecomunicações, e-com mercê e proteção de direitos autorais.

Ferguson et al., *Cryptography Engineering*

Muitos livros mostram como funcionam os algoritmos criptográficos populares. Isto livro explica como usar a criptografia - por que os protocolos criptográficos são desassinados do jeito que estão e como colocá-los juntos em um sistema que atenderá seus objetivos de segurança. É um livro bastante compacto, leitura essencial para qualquer um projeto de sistemas que dependem de criptografia.

Fridrich, *Esteganografia em mídia digital*

A esteganografia remonta à Grécia antiga, onde a cera foi derretida comprimidos em branco para que mensagens secretas pudessem ser aplicadas à madeira subjacente antes

a cera foi reaplicada. Hoje em dia, vídeos, áudio e outros conteúdos no Internet fornecem diferentes operadoras para mensagens secretas. Várias técnicas modernas para ocultar e localizar informações em imagens são discutidos aqui.

Página 908

884

LISTA DE LEITURA E BIBLIOGRAFIA

INDIVÍDUO. 9

Kaufman et al., *Network Security*, 2^a ed.

Este livro autoritário e espirituoso é o primeiro lugar para procurar por mais técnicas informações sobre algoritmos e protocolos de segurança de rede. Chave secreta e pública algoritmos e protocolos, hashes de mensagem, autenticação, Kerberos, PKI, IPsec, SSL / TLS e segurança de e-mail são explicados com cuidado e em considerável comprimento, com muitos exemplos. O Capítulo 26, sobre folclore de segurança, é uma verdadeira jóia. No

segurança, o diabo está nos detalhes. Qualquer pessoa que planeje projetar um sistema de segurança que realmente será usado aprenderá muito com os conselhos do mundo real neste capítulo ter.

Schneier, *segredos e mentiras*

Se você ler *Engenharia de criptografia* de capa a capa, você saberá tudo o que há para saber sobre algoritmos criptográficos. Se você então ler *Segredos e mentiras* de capa a capa (o que pode ser feito em muito menos tempo), você vai aprender que os algoritmos criptográficos não são tudo. Maior segurança fraquezas não são devidas a algoritmos defeituosos ou mesmo chaves que são muito curtas, mas a falhas no ambiente de segurança. Para uma discussão não técnica e fascinante de segurança de computador no sentido mais amplo, este livro é uma leitura muito boa.

Skoudis e Liston, *Counter Hack Reloaded*, 2^a ed.

A melhor maneira de parar um hacker é pensar como um hacker. Este livro mostra como hackers veem uma rede e argumentam que a segurança deve ser uma função de todo design da rede, não uma reflexão tardia baseada em uma tecnologia específica. Cobre quase todos os ataques comuns, incluindo os tipos de " engenharia social " que levam anúncios vantagem de usuários que nem sempre estão familiarizados com as medidas de segurança do computador.

9.2 BIBLIOGRAFIA ALFABÉTICA

ABRAMSON, N.: " Acesso à Internet usando VSATs," *IEEE Commun. Revista* , vol. 38, pp. 60–68, julho de 2000.

AHMADI, S.: " An Overview of Next-Generation Mobile WiMAX Technology," *IEEE Comun. Revista* , vol. 47, pp. 84-88, junho de 2009.

ALLMAN, M., e PAXSON, V.: " On Estimating End-to-End Network Path Properties," *Proc. SIGCOMM '99 Conf.* , ACM, pp. 263-274, 1999.

ANDERSON, C.: *The Long Tail: Why the Future of Business is Selling Less of More* , rev. atualiz. ed., New York: Hyperion, 2008a.

ANDERSON, RJ: *Engenharia de Segurança: Um Guia para Construir Confiável Distribuído Systems* , 2^a ed., Nova York: John Wiley & Sons, 2008b.

ANDERSON, RJ: " Free Speech Online and Offline," *IEEE Computer* , vol. 25, pp. 28-30, Junho de 2002.

Página 909

SEC. 9,2

BIBLIOGRAFIA ALFABÉTICA

885

- ANDERSON, RJ: "The Eternity Service," *Proc. Pragocrypt Conf.*, CTU Publishing House, pp. 242-252, 1996.
- ANDREWS, J., GHOSH, A., e MUHAMED, R.: *Fundamentals of WiMAX: Understanding Broadband Wireless Networking*, Upper Saddle River, NJ: Pearson Educação, 2007.
- ASSIM, D., DAHLMAN, E., FURUSKAR, A., JADING, Y., LINDSTROM, M., e PARKVALL, S.: "LTE: The Evolution of Mobile Broadband," *IEEE Commun. Magazine*, vol. 47, pp. 44-51, abril de 2009.
- BALLARDIE, T., FRANCIS, P., e CROWCROFT, J.: "Core Based Trees (CBT)," *Proc. SIGCOMM '93 Conf.*, ACM, pp. 85-95, 1993.
- BARAN, P.: "Sobre Comunicações Distribuídas: I. Introdução à Comunicação Distribuída Networks," *Memorandum RM-420-PR*, Rand Corporation, agosto de 1964.
- BELLAMY, J.: *Digital Telephony*, 3rd ed., New York: John Wiley & Sons, 2000.
- BELLMAN, RE: *Dynamic Programming*, Princeton, NJ: Princeton University Press, 1957.
- BELLOVIN, S.: "The Security Flag in the IPv4 Header," RFC 3514, abril de 2003.
- BELSNES, D.: "Flow Control in the Packet Switching Networks," *Communications Networks*, Uxbridge, England: Online, pp. 349-361, 1975.
- BENNET, CH e BRASSARD, G.: "Quantum Cryptography: Public Key Distribution and Coin Tossing," *Int'l Conf. on Computer Systems and Signal Processing*, pp. 175-179, 1984.
- BERESFORD, A., e STAJANO, F.: "Location Privacy in Pervasive Computing," *IEEE Pervasive Computing*, vol. 2, pp. 46-55, janeiro de 2003.
- BERGHEL, HL: "Cyber Privacy in the New Millennium," *IEEE Computer*, vol. 34, pp. 132-134, janeiro de 2001.
- BERNERS-LEE, T., CAILLIAU, A., LOUTONEN, A., NIELSEN, HF, e SECRET, A.: "The World Wide Web," *Commun. do ACM*, vol. 37, pp. 76-82, agosto de 1994.
- BERTSEKAS, D., e GALLAGER, R.: *Data Networks*, 2^a ed., Englewood Cliffs, NJ: Prentice Hall, 1992.
- BHATTI, SN e CROWCROFT, J.: "Fluxos Sensíveis de QoS: Problemas no Pacote IP Handling," *IEEE Internet Computing*, vol. 4, pp. 48-57, julho - agosto. 2000.
- BIHAM, E., e SHAMIR, A.: "Differential Fault Analysis of Secret Key Cryptosystems," *Proc. 17th Ann. Int'l Cryptology Conf.*, Berlin: Springer-Verlag LNCS 1294, pp. 513-525, 1997.
- BIRD, R., GOPAL, I., HERZBERG, A., JANSON, PA, KUTTEN, S., MOLVA, R., e YUNG, M.: "Projeto Sistemático de uma Família de Protetor de Autenticação Resistente a Ataques," *IEEE J. on Selected Areas in Commun.*, vol. 11, pp. 679-693, junho de 1993.
- BIRRELL, AD e NELSON, BJ: "Implementing Remote Procedure Calls," *ACM Trans. em Computer Systems*, vol. 2, pp. 39-59, fevereiro de 1984.

Página 910

886

LISTA DE LEITURA E BIBLIOGRAFIA

INDIVÍDUO. 9

- Biryukov, A., Shamir, A., e Wagner, D.: "Criptoanálise em tempo real de A5 / 1 em um PC," *Proc. Sétimo Workshop Internacional sobre Criptografia Rápida de Software*, Berlim: Springer-Verlag LNCS 1978, pp. 1-8, 2000.
- Blaze, M. e Bellovin, S.: "Tapping on My Network Door," *Commun. do ACM*, vol. 43, pág. 136, outubro de 2000.
- Booggs, D., Mogul, J., e Kent, C.: "Capacidade Medida de uma Ethernet: Mitos e Reality," *Proc. SIGCOMM '88 Conf.*, ACM, pp. 222-234, 1988.
- Borisov, N., Goldberg, I., e Wagner, D.: "Intercepting Mobile Communications: A Insegurança de 802.11," *Seventh Int'l Conf. em computação móvel e rede*, ACM, pp. 180-188, 2001.
- Braden, R.: "Requisitos para Hosts da Internet - Camadas de Comunicação," RFC 1122, Outubro de 1989.
- Braden, R., Borman, D., e Partridge, C.: "Computing the Internet Checksum," RFC 1071, setembro de 1988.
- Brandenburg, K.: "MP3 and AAC Explained," *Proc. 17th Intl. Conf.: alta qualidade Audio Coding*, Audio Engineering Society, pp. 99-110, agosto de 1999.
- Bray, T., Paoli, J., Spesberg-McQueen, C., Maler, E., Yergeau, F., e Cowan, J.: "Extensible Markup Language (XML) 1.1 (Segunda Edição)," W3C Recomendação, setembro de 2006.
- Breslau, L., CAO, P., Fan, L., Phillips, G., e Shenker, S.: "Web Caching and

- Zipf-like Distributions: Evidence and Implications," *Proc. INFOCOM Conf.* , IEEE, pp. 126–134, 1999.
- BURLEIGH, S., HOOKE, A., TORGERSON, L., FALL, K., CERF, V., DURST, B., SCOTT, K., e WEISS, H .:** " Rede Tolerante ao Delay: An Approach to Interplanetary Internet," *IEEE Commun. Revista* , vol. 41, pp. 128–136, junho de 2003.
- BURNETT, S., e PAINE, S .:** *RSA Security's Official Guide to Cryptography* , Berkeley, CA: Osborne / McGraw-Hill, 2001.
- BUSH, V.:** " As We May Think," *Atlantic Monthly* , vol. 176, pp. 101–108, julho de 1945.
- CAPETANAKIS, JI:** " Tree Algorithms for Packet Broadcast Channels," *IEEE Trans. em Teoria da Informação* , vol. IT – 5, pp. 505–515, setembro de 1979.
- CASTAGNOLI, G., BRAUER, S., e HERRMANN, M.:** " Optimization of Cyclic Redundancy Codes of verificação de dância com 24 e 32 bits de paridade," *IEEE Trans. em Commun.* , vol. 41, pp. 883-892, junho de 1993.
- CERF, V., e KAHN, R.:** " A Protocol for Packet Network Interconnection," *IEEE Trans. em Commun.* , vol. COM – 2, pp. 637–648, maio de 1974.
- CHANG, F., DEAN, J., GHEMAWAT, S., HSIEH, W., WALLACH, D., BURROWS, M., CHANDRA, T., FIKES, A., e GRUBER, R .:** " Bigtable: A Distributed Storage System para dados estruturados," *Proc. OSDI 2006 Symp.* , USENIX, pp. 15–29, 2006.
- CHASE, JS, GALLATIN, AJ, and YOCUM, KG:** " End System Optimizations for High-Velocidade TCP," *IEEE Commun. Revista* , vol. 39, pp. 68–75, abril de 2001.

Página 911

SEC. 9,2

BIBLIOGRAFIA ALFABÉTICA

887

- CHEN, S., e NAHRSTEDT, K .:** " Uma Visão Geral do Roteamento de QoS para Rede de Próxima Geração funciona," *IEEE Network Magazine* , vol. 12, pp. 64-69, novembro / dezembro. 1998.
- CHIU, D., e JAIN, R .:** " Análise dos Algoritmos de Aumento e Diminuição para Congestion Avoidance in Computer Networks," *Comput. Netw. ISDN Syst.* , vol. 17, pp. 1-4, Junho de 1989.
- CISCO:** " Cisco Visual Networking Index: Forecast and Methodology, 2009–2014," Cisco Systems Inc., junho de 2010.
- CLARK, DD:** " The Design Philosophy of the DARPA Internet Protocols," *Proc. SIGCOMM '88 Conf.* , ACM, pp. 106-114, 1988.
- CLARK, DD:** " Window and Acknowledgement Strategy in TCP," RFC 813, julho de 1982.
- CLARK, DD, JACOBSON, V., ROMKEY, J., e SALWEN, H .:** " An Analysis of TCP Processing Overhead," *IEEE Commun. Revista* , vol. 27, pp. 23-29, junho de 1989.
- CLARK, DD, SHENKER, S., e ZHANG, L .:** " Compatível com aplicativos em tempo real em um Rede de Pacotes de Serviços Integrados," *Proc. SIGCOMM '92 Conf.* , ACM, pp. 14-26, 1992.
- CLARKE, AC:** " Extra-Terrestrial Relays," *Wireless World* , 1945.
- CLARKE, I., MILLER, SG, HONG, TW, SANDBERG, O., e WILEY, B .:** " Protegendo Free Expression Online with Freenet," *IEEE Internet Computing* , vol. 6, pp. 40-49, Janeiro a fevereiro 2002
- COHEN, B .:** " Incentives Build Robustness in BitTorrent," *Proc. Primeiro Workshop sobre Economics of Peer-to-Peer Systems* , junho de 2003.
- COMER, DE:** *The Internet Book* , 4^a ed., Englewood Cliffs, NJ: Prentice Hall, 2007.
- COMER, DE:** *Internetworking with TCP / IP* , vol. 1, 5^a ed., Englewood Cliffs, NJ: Prentice Hall, 2005.
- CRAVER, SA, WU, M., LIU, B., STUBBLEFIELD, A., SWARTZLANDER, B., WALLACH, DW, DEAN, D., e FELTEN, EW:** " Lendo nas entrelinhas: lições do Desafio SDMI," *Proc. 10º USENIX Security Symp.* , USENIX, 2001.
- CROVELLA, M., e KRISHNAMURTHY, B .:** *Internet Measurement* , New York: John Wiley & Sons, 2006.
- DAEMEN, J., e RIJMEN, V .:** *The Design of Rijndael* , Berlin: Springer-Verlag, 2002.
- DALAL, Y., e METCLFE, R .:** " Encaminhamento de caminho reverso de pacotes de transmissão," *Commun. do ACM* , vol. 21, pp. 1040–1048, dezembro de 1978.
- DAVIE, B. e FARREL, A .:** *MPLS: Próximas etapas* , San Francisco: Morgan Kaufmann, 2008.
- DAVIE, B., and REKHTER, Y .:** *MPLS Technology and Applications* , San Francisco: Morgan Kaufmann, 2000.
- DAVIES, J .:** *Understanding IPv6* , 2nd ed., Redmond, WA: Microsoft Press, 2008.
- DAY, JD:** " The (Não) Revised OSI Reference Model," *Computer Commun. Rev.* , vol. 25, pp. 39–55, outubro de 1995.

Página 912

888

LISTA DE LEITURA E BIBLIOGRAFIA

INDIVÍDUO. 9

- DAY, JD e ZIMMERMANN, H.: "The OSI Reference Model," *Proc. do IEEE*, vol. 71, pp. 1334-1340, dezembro de 1983.
- DECANDIA, G., HASTORIN, D., JAMPANI, M., KAKULAPATI, G., LAKSHMAN, A., PIL-CHIN, A., SIVASUBRAMANIAN, S., VOSSHALL, P., e VOGELS, W.: "Dynamo: Armazenamento de valor-chave altamente disponível da Amazon," *Proc. 19th Symp. no sistema operacional tems Prin.*, ACM, pp. 205–220, dezembro de 2007.
- DEERING, SE: "SIP: Simple Internet Protocol," *IEEE Network Magazine*, vol. 7, pp. 16-28, maio / junho de 1993.
- DEERING, S., e CHERITON, D.: "Multicast Routing in Datagram Networks and Ex-LANs tendidas," *ACM Trans. em Computer Systems*, vol. 8, pp. 85-110, maio de 1990.
- DEMERS, A., KESHAV, S., e SHENKER, S.: "Analysis and Simulation of a Fair Queueing Algorithm," *Internetwok: Research and Experience*, vol. 1, pp. 3-26, setembro de 1990.
- DENNING, DE e SACCO, GM: "Timestamps in Key Distribution Protocols," *Comun. do ACM*, vol. 24, pp. 533-536, agosto de 1981.
- DEVARAPALLI, V., WAKIKAWA, R., PETRESCU, A., e THUBERT, P.: "Network Protocolo de suporte básico de mobilidade (NEMO)," RFC 3963, janeiro de 2005.
- DIFFIE, W., e HELLMAN, ME: "Exhaustive Cryptanalysis of the NBS Data Encryption Standard," *IEEE Computer*, vol. 10, pp. 74-84, junho de 1977.
- DIFFIE, W., and HELLMAN, ME: "New Directions in Cryptography," *IEEE Trans. em Teoria da Informação*, vol. IT – 2, pp. 644–654, novembro de 1976.
- DIJKSTRA, EW: "Uma nota sobre dois problemas em conexão com gráficos," *Numer. Matemática*, vol. 1, pp. 269-271, outubro de 1959.
- DILLEY, J., MAGGS, B., PARikh, J., PROKOP, H., SITARAMAN, R., e WHEIL, B.: "Globally Distributed Content Delivery," *IEEE Internet Computing*, vol. 6, pp. 50–58, 2002.
- DINGLEDINE, R., MATHEWSON, N., SYVERSON, P.: "Tor: The Second-Generation Onion Router," *Proc. 13º USENIX Security Symp.*, USENIX, pp. 303–320, agosto de 2004.
- DONAHOO, M., e CALVERT, K.: *TCP / IP Sockets in C*, 2ª ed., San Francisco: Morgan Kaufmann, 2009.
- DONAHOO, M., e CALVERT, K.: *TCP / IP Sockets in Java*, 2ª ed., San Francisco: Morgan Kaufmann, 2008.
- DONALDSON, G. e JONES, D.: "Cable Television Broadband Network Architectures," *IEEE Commun. Revista*, vol. 39, pp. 122-126, junho de 2001.
- DORFMAN, R.: "Detection of Defective Members of a Large Population," *Annals Math. Estatísticas*, vol. 14, pp. 436–440, 1943.
- DUTCHER, B.: *The NAT Handbook*, Nova York: John Wiley & Sons, 2001.
- DUTTA-ROY, A.: "Uma Visão Geral da Tecnologia de Modem a Cabo e Perspectivas de Mercado," *IEEE Commun. Revista*, vol. 39, pp. 81-88, junho de 2001.

Página 913

SEC. 9.2

BIBLIOGRAFIA ALFABÉTICA

889

- EDELMAN, B., OSTROVSKY, M., e SCHWARZ, M.: "Internet Advertising and the Geloilão de segundo preço eralizado: vendendo bilhões de dólares em palavras-chave," *American Economic Review*, vol. 97, pp. 242-259, março de 2007.
- EL GAMAL, T.: "A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logaritmos," *IEEE Trans. on Information Theory*, vol. IT – 1, pp. 469-472, julho de 1985.
- EPCGLOBAL: *Classe de Protocolos de Identidade de Radiofrequênciia EPC - Geração - RFID UHF Protocolo para comunicação de 860 MHz a 960 MHz Versão 1.2.0*, Bruxelas: EPCglobal Inc., outubro de 2008.
- FALL, K.: "A Delay-Tolerant Network Architecture for Challenged Internets," *Proc. SIGCOMM 2003 Conf.*, ACM, pp. 27-34, agosto de 2003.
- FALOUTSOS, M., FALOUTSOS, P., e FALOUTSOS, C.: "On Power-Law Relationships da Topologia da Internet," *Proc. SIGCOMM '99 Conf.*, ACM, pp. 251-262, 1999.
- FARRELL, S., e CAHILL, V.: *Delay- and Disruption-Tolerant Networking*, Londres: Artech House, 2007.
- FELLOWS, D., e JONES, D.: "DOCSIS Cable Modem Technology," *IEEE Commun. Revista*, vol. 39, pp. 202–209, março de 2001.
- FENNER, B., HANDLEY, M., HOLBROOK, H., e KOUVELAS, I.: "Protocol Independent Multicast-Sparse Mode (PIM-SM)," RFC 4601, agosto de 2006.

- FERGUSON, N., SCHNEIER, B., e KOHNO, T.** : *Engenharia de criptografia: Design Principles and Practical Applications*, Nova York: John Wiley & Sons, 2010.
- FLANAGAN, D.** : *JavaScript: The Definitive Guide*, 6^a ed., Sebastopol, CA: O'Reilly, 2010.
- FLETCHER, J.** : "An Arithmetic Checksum for Serial Transmissions," *IEEE Trans. em Comun.*, vol. COM – 0, pp. 247–252, janeiro de 1982.
- FLOYD, S., HANDLEY, M., PADHYE, J., e WIDMER, J.** : "Equation-Based Congestion Control for Unicast Applications," *Proc. SIGCOMM 2000 Conf.*, ACM, pp. 43–56, Agosto de 2000.
- FLOYD, S., e JACOBSON, V.** : "Random Early Detection for Congestion Avoidance," *IEEE / ACM Trans. on Networking*, vol. 1, pp. 397–413, agosto de 1993.
- FLUHRER, S., MANTIN, I., e SHAMIR, A.** : "Fraqueza na Programação Chave Algoritmo de RC4," *Proc. Oitava Ann. Workshop sobre áreas selecionadas em criptografia*, Berlin: Springer-Verlag LNCS 2259, pp. 1–24, 2001.
- FORD, B.** : "Streams estruturados: A New Transport Abstraction," *Proc. SIGCOMM 2007 Conf.*, ACM, pp. 361–372, 2007.
- FORD, LR, Jr., e FULKERSON, DR**: *Flows in Networks*, Princeton, NJ: Princeton University Press, 1962.
- FORD, W., e BAUM, MS**: *Secure Electronic Commerce*, Upper Saddle River, NJ: Prentice Hall, 2000.
- FORNEY, GD**: "The Viterbi Algorithm," *Proc. do IEEE*, vol. 61, pp. 268–278, março 1973.

Página 914

890

LISTA DE LEITURA E BIBLIOGRAFIA INDIVÍDUO. 9

- FOULI, K., e MALER, M.** : "The Road to Carrier-Grade Ethernet," *IEEE Commun. Revista*, vol. 47, pp. S30 – S38, março de 2009.
- FOX, A., GRIBBLE, S., BREWER, E., e AMIR, E.** : "Adaptando-se à rede e ao cliente Variabilidade via destilação dinâmica sob demanda," *SIGOPS Oper. Syst. Rev.*, vol. 30, pp. 160–170, dezembro de 1996.
- FRANCIS, P.** : "A Near-Term Architecture for Deploying Pip," *IEEE Network Magazine*, vol. 7, pp. 30–37, maio / junho de 1993.
- FRASER, AG**: "Towards a Universal Data Transport System," *IEEE J. on Selected Areas em Commun.*, vol. 5, pp. 803–816, novembro de 1983.
- FRIDRICH, J.** : *Steganography in Digital Media: Principles, Algorithms, and Applications*, Cambridge: Cambridge University Press, 2009.
- FULLER, V., e LI, T.** : "Classless Inter-domain Routing (CIDR): O endereço da Internet Assignment and Aggregation Plan," *RFC 4632*, agosto de 2006.
- GALLAGHER, RG**: "A Minimum Delay Routing Algorithm Using Distributed Computação," *IEEE Trans. em Commun.*, vol. COM – 5, pp. 73–85, janeiro de 1977.
- GALLAGHER, RG**: "Códigos de verificação de paridade de baixa densidade", *IRE Trans. na informação Teoria*, vol. 8, pp. 21–28, janeiro de 1962.
- GARFINKEL, S., com SPAFFORD, G.** : *Segurança da Web, Privacidade e Comércio*, Sebastopol, CA: O'Reilly, 2002.
- GAST, M.** : *802.11 Wireless Networks: The Definitive Guide*, 2^a ed., Sebastopol, CA: O'Reilly, 2005.
- GERSHENFELD, N., e KRIKORIAN, R., e COHEN, D.** : "A Internet das Coisas," *Scientific American*, vol. 291, pp. 76–81, outubro de 2004.
- GILDER, G.** : "Metcalf's Law and Legacy," *Forbes ASAP*, Sepy. 13, 1993.
- GOODE, B.** : "Voice over Internet Protocol," *Proc. do IEEE*, vol. 90, pp. 1495–1517, Setembro de 2002.
- GORALSKI, WJ**: *SONET*, 2^a ed., Nova York: McGraw-Hill, 2002.
- GRAYSON, M., SHATZKAMER, K., e WAINER, S.** : *IP Design for Mobile Networks*, Indianápolis, IN: Cisco Press, 2009.
- GROBE, K., e ELBERS, J.** : "PON in Adolescence: From TDMA to WDM-PON," *IEEE Comun. Revista*, vol. 46, pp. 26–34, janeiro de 2008.
- GROSS, G., KAYCEE, M., LIN, A., MALIS, A., e STEPHENS, J.** : "O PPP acabou AAL5," *RFC 2364*, julho de 1998.
- HA, S., RHEE, I., e LISONG, X.** : "CUBIC: A New TCP-Friendly High-Speed TCP Vari-formiga," *SIGOPS Oper. Syst. Rev.*, vol. 42, pp. 64–74, junho de 2008.
- HAFNER, K., e LYON, M.** : *Where Wizards Stay Up Late*, New York: Simon & Schuster, 1998.
- HALPERIN, D., HEYDT-BENJAMIN, T., RANSFORD, B., CLARK, S., DEFEND, B., MORGAN, W., FU, K., KOHNO, T., e MAISEL, W.** : "Pacemakers and Implantable Cardi-

SEC. 9,2

BIBLIOGRAFIA ALFABÉTICA

891

- Desfibriladores ac: Software Radio Attacks and Zero-Power Defenses, " *IEEE Symp. em Security and Privacy* , pp. 129-142, maio de 2008.
- HALPERIN, D., HU, W., SHETH, A. e WETHERALL, D. :** " 802.11 with Multiple Antennas for Dummies, " *Computer Commun. Rev.* , vol. 40, pp. 19-25, janeiro de 2010.
- HAMMING, RW:** " Códigos de detecção e correção de erros, " *Bell System Tech. J.* , vol. 29, pp. 147-160, abril de 1950.
- HARTE, L., KELLOGG, S., DREHER, R. e SCHAFFNIT, T. :** *The Comprehensive Guide to Wireless Technology* , Fuquay-Varina, NC: APDG Publishing, 2000.
- HAWLEY, GT:** " Perspectivas históricas no circuito telefônico dos EUA ", *IEEE Commun. Revista* , vol. 29, pp. 24-28, março de 1991.
- HECHT, J. :** *Understanding Fiber Optics* , Upper Saddle River, NJ: Prentice Hall, 2005.
- HELD, G. :** *A Practical Guide to Content Delivery Networks* , 2^a ed., Boca Raton, FL: CRC Press, 2010.
- HEUSSE, M., ROUSSEAU, F., BERGER-SABBATEL, G., DUDA, A. :** " Performance Anomaly de 802.11b, " *Proc. INFOCOM Conf.* , IEEE, pp. 836-843, 2003.
- HIERTZ, G., DENTENEER, D., STIBOR, L., ZANG, Y., COSTA, X., e WALKE, B. :** " O IEEE 802.11 Universe, " *IEEE Commun. Revista* , vol. 48, pp. 62-70, janeiro de 2010.
- HOE, J. :** " Melhorando o comportamento inicial de um esquema de controle de congestionamento para TCP, " *Proc. SIGCOMM '96 Conf.* , ACM, pp. 270-280, 1996.
- HU, Y. e LI, VOK:** " Internet baseada em satélite: um tutorial, " *IEEE Commun. Revista* , vol. 30, pp. 154-162, março de 2001.
- HUIITEMA, C. :** *Routing in the Internet* , 2^a ed., Englewood Cliffs, NJ: Prentice Hall, 1999.
- HULL, B., BYCHKOVSKY, V., CHEN, K., GORACZKO, M., MIU, A., SHIH, E., ZHANG, Y., BALAKRISHNAN, H., e MADDEN, S. :** " CarTel: A Distributed Mobile Sensor Computing System, " *Proc. Sensys 2006 Conf.* , ACM, pp. 125-138, novembro de 2006.
- HUNTER, D., RAFTER, J., FAWCETT, J., VAN DER LIST, E., AYERS, D., DUCKETT, J., WATT, A. e MCKINNON, L. :** *Beginning XML* , 4^a ed., New Jersey: Wrox, 2007.
- IRMER, T. :** " Shaping Future Telecommunications: The Challenge of Global Standardização, " *IEEE Commun. Revista* , vol. 32, pp. 20-28, janeiro de 1994.
- ITU (INTERNATIONAL TELECOMMUNICATION UNION):** *ITU Internet Reports 2005: The Internet of Things* , Genebra: ITU, novembro de 2005.
- ITU (INTERNATIONAL TELECOMMUNICATION UNION):** *Medindo as Informações Sociedade: The ICT Development Index* , Genebra: ITU, março de 2009.
- JACOBSON, V. :** " Compressing TCP / IP Headers for Low-Speed Serial Links, " RFC 1144, Fevereiro de 1990.
- JACOBSON, V. :** " Congestion Avoidance and Control, " *Proc. SIGCOMM '88 Conf.* , ACM, pp. 314-329, 1988.

892

LISTA DE LEITURA E BIBLIOGRAFIA

INDIVÍDUO. 9

- JAIN, R., e ROUTHIER, S. :** " Pacotes de Trens - Medições e um Novo Modelo para Comtráfego de rede de computador, " *IEEE J. on Selected Areas in Commun.* , vol. 6, pp. 986-995, Setembro de 1986.
- JAKOBSSON, M., e WETZEL, S. :** " Security Weaknesses in Bluetooth, " *Topics in Cryptologia: CT-RSA 2001* , Berlin: Springer-Verlag LNCS 2020, pp. 176-191, 2001.
- JOEL, A. :** " Telecomunicações e a IEEE Communications Society, " *IEEE Commun. Magazine* , Edição do 50º Aniversário, pp. 6-14 e 162-167, maio de 2002.
- JOHNSON, D., PERKINS, C., e ARKKO, J. :** " Mobility Support in IPv6, " RFC 3775, Junho de 2004.
- JOHNSON, DB, MALTZ, D., e BROCH, J. :** " DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks, " *Ad Hoc Networking* , Boston: Addison-Wesley, pp. 139-172, 2001.
- JUANG, P., OKI, H., WANG, Y., MARTONOSI, M., PEH, L., e RUBENSTEIN, D. :** " Economia de energia para rastreamento de vida selvagem: compensações de design e experiências iniciais com a ZebraNet, " *SIGOPS Oper. Syst. Rev.* , vol. 36, pp. 96-107, outubro de 2002.

- KAHN, D.** : *The Codebreakers*, 2^a ed., New York: Macmillan, 1995.
- KAMOUN, F., e KLEINROCK, L.** : " Avaliação de Desempenho Estocástico Hierárquico Routing for Large Networks," *Computer Networks*, vol. 3, pp. 337-353, novembro de 1979.
- KARN, P.** : " MACA — A New Channel Access Protocol for Packet Radio," *ARRL / CRRL Rádio Amador Nona Rede de Computadores Conf.*, pp. 134-140, 1990.
- KARN, P., e PARTRIDGE, C.** : " Melhorando as estimativas de ida e volta em transporte confiável Protocolos," *Proc. SIGCOMM '87 Conf.*, ACM, pp. 2-7, 1987.
- KARP, B. e KUNG, HT.** : " GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," *Proc. MOBICOM 2000 Conf.*, ACM, pp. 243-254, 2000.
- KASIM, A.** : *Delivering Carrier Ethernet*, New York: McGraw-Hill, 2007.
- KATABI, D., HANDLEY, M., e ROHRS, C.** : " Internet Congestion Control for Future High Bandwidth-Delay Product Environments," *Proc. SIGCOMM 2002 Conf.*, ACM, pp. 89-102, 2002.
- KATZ, D. e FORD, PS.** : " TUBA: Substituindo IP por CLNP," *Revista IEEE Network*, vol. 7, pp. 38-47, maio / junho de 1993.
- KAUFMAN, C., PERLMAN, R., e SPECINER, M.** : *Network Security*, 2^a ed., Englewood Cliffs, NJ: Prentice Hall, 2002.
- KENT, C. e MOGUL, J.** : " Fragmentation Considered Harmful," *Proc. SIGCOMM '87 Conf.*, ACM, pp. 390-401, 1987.
- KERCKHOFF, A.** : " La Cryptographie Militaire," *J. des Sciences Militaires*, vol. 9, pp. 5-38, janeiro de 1883 e pp. 161-191, fevereiro de 1883.
- KHANNA, A., e ZINKY, J.** : " The Revised ARPANET Routing Metric," *Proc. SIGCOMM '89 Conf.*, ACM, pp. 45-56, 1989.
- KIPNIS, J.** : " Batendo o Sistema: Abusos do Processo de Adoção de Padrões," *IEEE Commun. Revista*, vol. 38, pp. 102-105, julho de 2000.

Página 917

SEC. 9,2

BIBLIOGRAFIA ALFABÉTICA

893

- KLEINROCK, L.** : " Poder e outras regras determinísticas de ouro para Probabilistic Problems in Computer Communications," *Proc. Intl. Conf. em Commun.*, pp. 43.1.1-43.1.10, junho de 1979.
- KLEINROCK, L., e TOBAGI, F.** : " Técnicas de acesso aleatório para transmissão de dados sobre Canais de Rádio Comutados por Pacote," *Proc. Nat. Computer Conf.*, pp. 187-201, 1975.
- KOHLER, E., HANDLEY, H., e FLOYD, S.** : " Projetando DCCP: Controle de Congestionamento sem confiabilidade," *Proc. SIGCOMM 2006 Conf.*, ACM, pp. 27-38, 2006.
- KOODLI, R. e PERKINS, CE.** : *Mobile Inter-networking with IPv6*, New York: John Wiley & Sons, 2007.
- KOOPMAN, P.** : "32 -Bit Cyclic Redundancy Codes for Internet Applications," *Proc. Intl. Conf. em sistemas e redes confiáveis.*, IEEE, pp. 459-472, 2002.
- KRISHNAMURTHY, B., e REXFORD, J.** : *Web Protocols and Practice*, Boston: Addison-Wesley, 2001.
- KUMAR, S., PAAR, C., PELZL, J., PFEIFFER, G., e SCHIMMELER, M.** : " Quebra Ciphers with COPACOBANA: A Cost-Optimized Parallel Code Breaker," *Proc. 8º Hardware criptográfico e sistemas incorporados Wksp.*, IACR, pp. 101-118, outubro 2006.
- LABOVITZ, C., AHUJA, A., BOSE, A., and JAHANIAN, F.** : " Delayed Internet Routing Convergence," *IEEE / ACM Trans. on Networking*, vol. 9, pp. 293-306, junho de 2001.
- LAM, CKM e TAN, BCY.** : " The Internet Is Changing the Music Industry," *Commun. do ACM*, vol. 44, pp. 62-66, agosto de 2001.
- LAOUTARIS, N., SMARAGDAKIS, G., RODRIGUEZ, P., e SUNDARAM, R.** : " Atraso Tolerant Bulk Data Transfers on the Internet," *Proc. SIGMETRICS 2009 Conf.*, ACM, pp. 229-238, junho de 2009.
- LARMO, A., LINDSTROM, M., MEYER, M., PELLETIER, G., TORSNER, J., e WIEMANN, H.** : " The LTE Link-Layer Design," *IEEE Commun. Revista*, vol. 47, pp. 52-59, abril de 2009.
- LEE, JS e MILLER, LE.** : *CDMA Systems Engineering Handbook*, Londres: Artech House, 1998.
- LELAND, W., TAQQU, M., WILLINGER, W., e WILSON, D.** : " On the Self-Similar Nature of Ethernet Traffic," *IEEE / ACM Trans. on Networking*, vol. 2, pp. 1-15, fevereiro 1994.
- LEMON, J.** : " Resisting SYN Flood DOS Attacks com um SYN Cache," *Proc. BSDCon Conf.*, USENIX, pp. 88-98, 2002.
- LEVY, S.** : " Crypto Rebels," *Wired*, pp. 54-61, maio / junho de 1993.

- LEWIS, M.** : *Comparing, Designing and Deploying VPNs* , Indianapolis, IN: Cisco Press, 2006.
- LI, M., AGRAWAL, D., GANESAN, D., e VENKATARAMANI, A.** : " Block-Switched Networks: A New Paradigm for Wireless Transport, " *Proc. NSDI 2009 Conf.* , USENIX, pp. 423–436, 2009.

Página 918

894

LISTA DE LEITURA E BIBLIOGRAFIA

INDIVÍDUO. 9

- LIN, S., e COSTELLO, D.** : *Error Control Coding* , 2^a ed., Upper Saddle River, NJ: Pearson Education, 2004.
- LUBACZ, J., MAZURCZYK, W., e SZCZYPiorski, K.** : " Vice over IP, " *IEEE Spec-trum* , pp. 42-47, fevereiro de 2010.
- Macedonia, MR.** : " Distributed File Sharing, " *IEEE Computer* , vol. 33, pp. 99-101, 2000.
- MADHAVAN, J., KO, D., LOT, L., GANGPATHY, V., RASMUSSEN, A., e HALEVY, A.** : " Deep Web Crawl do Google, " *Proc. VLDB 2008 Conf.* , VLDB Endowment, pp. 1241–1252, 2008.
- MAHAJAN, R., RODRIG, M., WETHERALL, D., e ZAHORJAN, J.** : " Analisando o MAC-Level Behavior of Wireless Networks in the Wild, " *Proc. SIGCOMM 2006 Conf.* , ACM, pp. 75–86, 2006.
- MALIS, A., e SIMPSON, W.** : " PPP over SONET / SDH, " RFC 2615, junho de 1999.
- MASSEY, JL.** : " Shift-Register Synthesis and BCH Decoding, " *IEEE Trans. no Infor-maria da mação* , vol. IT – 5, pp. 122–127, janeiro de 1969.
- MATSUI, M.** : " Linear Cryptanalysis Method for DES Cipher, " *Advances in Cryptology—Eurocrypt 1993 Proceedings* , Berlin: Springer-Verlag LNCS 765, pp. 386-397, 1994.
- MAUFER, TA.** : *IP Fundamentals* , Upper Saddle River, NJ: Prentice Hall, 1999.
- MAYMOUNKOV, P., e MAZIERES, D.** : " Kademia: A Peer-to-Peer Information System Com base no XOR Metric, " *Proc. First Intl. Wksp. on Peer-to-Peer Systems* , Berlim: Springer-Verlag LNCS 2429, pp. 53-65, 2002.
- MAZIERES, D., e KAASHOEK, MF.** : " O Projeto, Implementação e Operação de an Email Pseudonym Server, " *Proc. Fifth Conf. no computador e no Commun. Segurança* , ACM, pp. 27-36, 1998.
- MCAFEE LABS:** *McAfee Threat Reports: First Quarter 2010* , McAfee Inc., 2010.
- MENEZES, AJ, e VANSTONE, SA.** : " Elliptic Curve Cryptosystems and their Imple-mentação, " *Journal of Cryptology* , vol. 6, pp. 209-224, 1993.
- MERKLE, RC e HELLMAN, M.** : " Escondendo e Assinaturas em Mochilas do Alça-pão, " *IEEE Trans. on Information Theory* , vol. IT – 4, pp. 525–530, setembro de 1978.
- METCALFE, RM.** : " Computer / Network Interface Design: Lessons from Arpanet and Ethernet, " *IEEE J. on Selected Areas in Commun.* , vol. 11, pp. 173-179, fevereiro de 1993.
- METCALFE, RM e BOGGS, DR.** : " Ethernet: Comutação de pacotes distribuídos para local Redes de Computadores, " *Commun. do ACM* , vol. 19, pp. 395–404, julho de 1976.
- METZ, C.** : " Interconnecting ISP Networks, " *IEEE Internet Computing* , vol. 5, pp. 74-80, Março a abril 2001.
- MISHRA, PP, KANAKIA, H., e TRIPATHI, S.** : " On Hop por Hop Rate-Based Conges-de controle, " *IEEE / ACM Trans. on Networking* , vol. 4, pp. 224-239, abril de 1996.
- MOGUL, JC.** : " IP Network Performance, " in *Internet System Handbook* , DC Lynch e MY Rose (eds.), Boston: Addison-Wesley, pp. 575–575, 1993.

Página 919

SEC. 9,2

BIBLIOGRAFIA ALFABÉTICA

895

MOGUL, J., e DEERING, S. : " Path MTU Discovery, " RFC 1191, novembro de 1990.

MOGUL, J., e MINSHALL, G. : " Rethinking the Nagle Algorithm, " *Comput. Commun. Rev.* , vol. 31, pp. 6-20, janeiro de 2001.

MOY, J. : " Multicast Routing Extensions for OSPF, " *Commun. do ACM* , vol. 37, pp. 61-66, agosto de 1994.

MULLINS, J. : " Making Unbreakable Code, " *IEEE Spectrum* , pp. 40-45, maio de 2002.

NAGLE, J. : " On Packet Switches with Infinite Storage, " *IEEE Trans. em Commun.* , vol. COM – 5, pp. 435–438, abril de 1987.

NAGLE, J. : " Congestion Control in TCP / IP Internetworks, " *Computer Commun. Rev.* , vol. 14, pp. 11-17, outubro de 1984.

- NAUGHTON, J.** : *A Brief History of the Future*, Woodstock, NY: Overlook Press, 2000.
- NEEDHAM, RM e SCHROEDER, MD:** " Usando criptografia para autenticação em Grandes Redes de Computadores," *Commun. do ACM*, vol. 21, pp. 993–999, dez. 1978.
- NEEDHAM, RM e SCHROEDER, MD:** " Authentication Revisited," *Operating Systems Rev.*, vol. 21, pág. 7 de janeiro de 1987.
- NELAKUDITI, S., e ZHANG, Z.-L.:** " A Localized Adaptive Proportioning Approach to Roteamento QoS," *IEEE Commun. Revista* vol. 40, pp. 66-71, junho de 2002.
- NEUMAN, C., and TS' O, T.:** " Kerberos: An Authentication Service for Computer Networks," *IEEE Commun. Mag.*, vol. 32, pp. 33-38, setembro de 1994.
- NICHOLS, RK e LEKKAS, PC:** *Wireless Security*, Nova York: McGraw-Hill, 2002.
- NIST:** " Secure Hash Algorithm," US Government Federal Information Processing Standard 180, 1993.
- NONNENMACHER, J., BIERSACK, E., e TOWSLEY, D.:** " Recuperação de perda baseada em paridade for Reliable Multicast Transmission," *Proc. SIGCOMM '97 Conf.*, ACM, pp. 289–300, 1997.
- NUCCI, A., e PAPAGIANNAKI, D.:** *Projeto, Medição e Gestão de Grandes Scale IP Networks*, Cambridge: Cambridge University Press, 2008.
- NUGENT, R., MUNAKANA, R., CHIN, A., COELHO, R., e PUIG-SUARI, J.:** " O CubeSat: O Padrão Picoataforma para Pesquisa e Educação," *Proc. ESPAÇO 2008 Conf.*, AIAA, 2008.
- ORAN, D.:** " OSI IS-IS Intra-domain Routing Protocol," RFC 1142, fevereiro de 1990.
- OTWAY, D., e REES, O.:** " Autenticação mútua eficiente e oportunista," *Operacional Systems Rev.*, pp. 8–10, janeiro de 1987.
- PADHYE, J., FIROIU, V., TOWSLEY, D., e KUROSE, J.:** " Modeling TCP Throughput: Um modelo simples e sua validação empírica," *Proc. SIGCOMM '98 Conf.*, ACM, pp. 303-314, 1998.
- PALAIS, JC:** *Fiber Optic Commun.*, 5^a ed., Englewood Cliffs, NJ: Prentice Hall, 2004.

Página 920

896

LISTA DE LEITURA E BIBLIOGRAFIA

INDIVÍDUO. 9

- PARAMESWARAN, M., SUSARLA, A., e WHINSTON, AB:** " Rede P2P: An Information-Sharing Alternative," *IEEE Computer*, vol. 34, pp. 31-38, julho de 2001.
- PAREKH, A., e GALLAGHER, R.:** " Uma Abordagem Generalizada de Compartilhamento de Processador para Fluxo Controle em redes de serviços integrados: o caso de vários nós," *IEEE / ACM Trans. on Networking*, vol. 2, pp. 137-150, abril de 1994.
- PAREKH, A., e GALLAGHER, R.:** " Uma Abordagem Generalizada de Compartilhamento de Processador para Fluxo Controle em redes de serviços integrados: O caso de nó único," *IEEE / ACM Trans. on Networking*, vol. 1, pp. 344-357, junho de 1993.
- PARTRIDGE, C., HUGHES, J., e STONE, J.:** " Performance of Checksums and CRCs sobre dados reais," *Proc. SIGCOMM '95 Conf.*, ACM, pp. 68-76, 1995.
- PARTRIDGE, C., MENDEZ, T., e MILLIKEN, W.:** " Host Anycasting Service," RFC 1546, novembro de 1993.
- PAXSON, V., e FLOYD, S.:** " Wide-Area Traffic: The Failure of Poisson Modeling," *IEEE / ACM Trans. on Networking*, vol. 3, pp. 226-244, junho de 1995.
- PERKINS, C.:** " IP Mobility Support for IPv4," RFC 3344, agosto de 2002.
- PERKINS, CE:** *RTP: Áudio e Vídeo para a Internet*, Boston: Addison-Wesley, 2003.
- PERKINS, CE (ed.):** *Ad Hoc Networking*, Boston: Addison-Wesley, 2001.
- PERKINS, CE:** *Princípios e práticas de design de IP móvel*, Upper Saddle River, NJ: Prentice Hall, 1998.
- PERKINS, CE, e ROYER, E.:** " O protocolo Ad Hoc On-Demand Distance-Vector," em *Ad Hoc Networking*, editado por C. Perkins, Boston: Addison-Wesley, 2001.
- PERLMAN, R.:** *Interconexões*, 2^a ed., Boston: Addison-Wesley, 2000.
- PERLMAN, R.:** *Network Layer Protocols with Byzantine Robustness*, Ph.D. tese, MIT, 1988.
- PERLMAN, R.:** " Um algoritmo para a computação distribuída de uma árvore de expansão em um LAN estendida," *Proc. SIGCOMM '85 Conf.*, ACM, pp. 44-53, 1985.
- PERLMAN, R., e KAUFMAN, C.:** " Key Exchange in IPsec," *IEEE Internet Computing*, vol. 4, pp. 50–56, novembro – dezembro. 2000.
- PETERSON, WW e BROWN, DT:** " Cyclic Codes for Error Detection," *Proc. IRE*, vol. 49, pp. 228–235, janeiro de 1961.
- PIATEK, M., KOHNO, T., e KRISHNAMURTHY, A.:** " Desafios e orientações para

Monitorando redes de compartilhamento de arquivos P2P - ou por que minha impressora recebeu um DMCA Aviso de remoção," 3º workshop sobre tópicos importantes em segurança , USENIX, julho de 2008.

PIATEK, M., ISDAL, T., ANDERSON, T., KRISHNAMURTHY, A. e VENKA-TARAMANI, V.: " Os incentivos criam robustez no BitTorrent ?, " Proc. INDE 2007 Conf. , USENIX, pp. 1-14, 2007.

PISCITELLO, DM e CHAPIN, AL: *Rede de sistemas abertos: TCP / IP e OSI* , Boston: Addison-Wesley, 1993.

Página 921

SEC. 9,2

BIBLIOGRAFIA ALFABÉTICA

897

- PIVA, A., BARTOLINI, F., e BARNI, M.:** " Gerenciando Direitos Autorais em Redes Abertas, " *IEEE Internet Computing* , vol. 6, pp. 18–26, maio– 2002.
- POSTEL, J.:** " Internet Control Message Protocols, " RFC 792, setembro de 1981.
- RABIN, J., e M c CATHIENEVILE, C.:** " Mobile Web Best Practices 1.0, " W3C Recomendação, julho de 2008.
- RAMAKRISHNAM, KK, FLOYD, S., e BLACK, D.:** " The Addition of Explicit Congestion-Notificação de notificação (ECN) para IP, " RFC 3168, setembro de 2001.
- RAMAKRISHNAN, KK e JAIN, R.:** " Um esquema de feedback binário para congestionamento Avoidance in Computer Networks with a Connectionless Network Layer, " *Proc. SIGCOMM '88 Conf.* , ACM, pp. 303-313, 1988.
- RAMASWAMI, R., KUMAR, S., and SASAKI, G.:** *Optical Networks: A Practical Perspective* , 3ª ed., San Francisco: Morgan Kaufmann, 2009.
- RATNASAMY, S., FRANCIS, P., HANDLEY, M., KARP, R., e SHENKER, S.:** " A Scal-Rede Endereçável por Conteúdo, " *Proc. SIGCOMM 2001 Conf.* , ACM, pp. 161–172, 2001.
- RIEBACK, M., CRISPO, B., e TANENBAUM, A.:** " Seu gato está infectado com um Com-puter Virus ?, " *Proc. IEEE Percom* , pp. 169–179, março de 2006.
- RIVEST, RL:** " The MD5 Message-Digest Algorithm, " RFC 1320, abril de 1992.
- RIVEST, RL, SHAMIR, A., e ADLEMAN, L.:** " On a Method for Obtendo Digital Sig-naturezas e criptosistemas de chave pública, " *Commun. do ACM* , vol. 21, pp. 120-126, Fevereiro de 1978.
- ROBERTS, LG:** " Extensions of Packet Communication Technology to a Hand Held Per-Terminal sonal, " *Proc. Spring Joint Computer Conf.* , AFIPS, pp. 295–298, 1972.
- ROBERTS, LG:** " Multiple Computer Networks and Intercomputer Communication, " *Proc. First Symp. em sistemas operacionais Prin.* , ACM, pp. 3.1-3.6, 1967.
- ROSE, MT:** *The Simple Book* , Englewood Cliffs, NJ: Prentice Hall, 1994.
- ROSE, MT:** *The Internet Message* , Englewood Cliffs, NJ: Prentice Hall, 1993.
- ROWSTRON, A., e DRUSCHEL, P.:** " Pastelaria: Escalável, Localização de Objetos Distribuídos e Routing for Large Scale Peer-to-Peer Storage Utility, " *Proc. 18th Int'l Conf. em Dis-Tributed Systems Platforms* , Londres: Springer-Verlag LNCS 2218, pp. 329-350, 2001.
- RUIZ-SANCHEZ, MA, BIERSACK, EW e DABBOUS, W.:** " Levantamento e Taxonomia de IP Address Lookup Algorithms, " *IEEE Network Magazine* , vol. 15, pp. 8-23, Março a abril 2001.
- SALTZER, JH, REED, DP e CLARK, DD:** " Argumentos de ponta a ponta no sistema de-sinal, " *ACM Trans. em Computer Systems* , vol. 2, pp. 277-288, novembro de 1984.
- SAMPLE, A., YEAGER, D., POWLEDGE, P., MAMISHEV, A., e SMITH, J.:** " Design of uma plataforma de detecção programável sem bateria baseada em RFID, " *IEEE Trans. em Instrumentation and Measurement* , vol. 57, pp. 2608–2615, novembro de 2008.
- SAROIU, S., GUMMADI, K., e GRIBBLE, S.:** " Measuring and Analyzing the Charac-terísticas do Napster e do Gnutella Hosts, " *Multim. Syst.* , vol. 9 , pp. 170-184, agosto de 2003.

Página 922

898

LISTA DE LEITURA E BIBLIOGRAFIA

INDIVÍDUO. 9

- SCHALLER, R.:** " Lei de Moore: Passado, Presente e Futuro, " *IEEE Spectrum* , vol. 34, pp. 52–59, junho de 1997.
- SCHNEIER, B.:** *Secrets and Lies* , Nova York: John Wiley & Sons, 2004.
- SCHNEIER, B.:** *E-Mail Security* , Nova York: John Wiley & Sons, 1995.
- SCHNORR, CP:** " Efficient Signature Generation for Smart Cards, " *Journal of Cryptology* , vol. 4, pp. 161-174, 1991.
- SCHOLTZ, RA:** " The Origins of Spread-Spectrum Communications, " *IEEE Trans. em*

- Comum.*, vol. COM – 0, pp. 822–854, maio de 1982.
- SCHWARTZ, M., e ABRAMSON, N.:** "The AlohaNet: Surfing for Wireless Data," *IEEE Comum. Revista*, vol. 47, pp. 21-25, dezembro de 2009.
- SEIFERT, R. e EDWARDS, J.:** *The All-New Switch Book*, NY: John Wiley, 2008.
- SENN, JA.:** "The Emergence of M-Commerce," *IEEE Computer*, vol. 33, pp. 148-150, Dezembro de 2000.
- SERJANTOV, A.:** "Anonymizing Censorship Resistant Systems," *Proc. Primeiro Internacional Workshop on Peer-to-Peer Systems*, Londres: Springer-Verlag LNCS 2429, pp. 111-120, 2002.
- SHACHAM, N., e MCKENNY, P.:** "Recuperação de pacotes em redes de alta velocidade usando Coding and Buffer Management," *Proc. INFOCOM Conf.*, IEEE, pp. 124-131, junho 1990.
- SHAIKH, A., REXFORD, J., e SHIN, K.:** "Load-Sensitive Routing of Long-Lived IP Flows," *Proc. SIGCOMM '99 Conf.*, ACM, pp. 215-226, setembro de 1999.
- SHALUNOV, S., e CARLSON, R.:** "Detecting Duplex Mismatch on Ethernet," *Passive and Active Network Measurement*, Berlin: Springer-Verlag LNCS 3431, pp. 3135-3148, 2005.
- SHANNON, C.:** "A Mathematical Theory of Communication," *Bell System Tech. J.*, vol. 27, pp. 379-423, julho de 1948; e pp. 623–656, outubro de 1948.
- SHEPARD, S.:** *SONET / SDH Demystified*, New York: McGraw-Hill, 2001.
- SHREEDHAR, M., and VARGHESE, G.:** "Efficient Fair Queuing using Deficit Round Robin," *Proc. SIGCOMM '95 Conf.*, ACM, pp. 231-243, 1995.
- SIMPSON, W.:** *Video Over IP*, 2^a ed., Burlington, MA: Focal Press, 2008.
- SIMPSON, W.:** "PPP in HDLC-like Framing," RFC 1662, July 1994b.
- SIMPSON, W.:** "The Point-to-Point Protocol (PPP)," RFC 1661, July 1994a.
- SIU, K., e JAIN, R.:** "Uma Breve Visão Geral do ATM: Camadas de Protocolo, Emulação de LAN e Traffic," *ACM Computer Communications Review*, vol. 25, pp. 6–20, abril de 1995.
- SKOUDIS, E., e LISTON, T.:** *Counter Hack Reloaded*, 2^a ed., Upper Saddle River, NJ: Prentice Hall, 2006.
- SMITH, DK e ALEXANDER, RC:** *Fumbling the Future*, Nova York: William Mor-Row, 1988.

Página 923

SEC. 9,2

BIBLIOGRAFIA ALFABÉTICA

899

- SNOEREN, AC, e BALAKRISHNAN, H.:** "Uma Abordagem de Ponta a Ponta para Host Mobility," *Int'l Conf. on Mobile Computing and Networking*, ACM, pp. 155-166, 2000.
- SOBEL, DL:** "Will Carnivore Devour Online Privacy," *IEEE Computer*, vol. 34, pp. 87-88, maio de 2001.
- SOTIROV, A., STEVENS, M., APPELBAUM, J., LENSTRA, A., MOLNAR, D., OSVIK, D., e DE WEGER, B.:** "MD5 Considered Harmful Today," *Proc. 25º Chaos Commu-Congresso de nicação*, Verlag Art d'Ameublement, 2008.
- SOUTHEY, R.:** *The Doctors*, Londres: Longman, Brown, Green and Longmans, 1848.
- SPURGEON, CE:** *Ethernet: The Definitive Guide*, Sebastopol, CA: O'Reilly, 2000.
- STALLINGS, W.:** *Data and Computer Communications*, 9^a ed., Upper Saddle River, NJ: Pearson Education, 2010.
- STARR, T., SORBARA, M., COIFFI, J., e SILVERMAN, P.:** "DSL Advances," Upper Saddle River, NJ: Prentice Hall, 2003.
- STEVENS, WR:** *TCP / IP Illustrated: The Protocols*, Boston: Addison Wesley, 1994.
- STINSON, DR:** *Cryptography Theory and Practice*, 2^a ed., Boca Raton, FL: CRC Press, 2002
- STOICA, I., MORRIS, R., KARGER, D., KAASHOEK, MF, e BALAKRISHNAN, H.:** "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications," *Proc. SIGCOMM 2001 Conf.*, ACM, pp. 149-160, 2001.
- STUBBLEFIELD, A., IOANNIDIS, J., e RUBIN, AD:** "Using the Fluhrer, Mantin, and Shamir Attack to Break WEP," *Proc. Segurança de rede e sistemas distribuídos Symp.*, ISOC, pp. 1-11, 2002.
- STUTTARD, D., e PINTO, M.:** *The Web Application Hacker's Handbook*, Nova York: John Wiley & Sons, 2007.
- SU, S.:** *The UMTS Air Interface in RF Engineering*, New York: McGraw-Hill, 2007.
- SULLIVAN, G., e WIEGAND, T.:** "Tree Algorithms for Packet Broadcast Channels," *Proc. do IEEE*, vol. 93, pp. 18-31, janeiro de 2005.
- SUNSHINE, CA e DALAL, YK:** "Connection Management in Transport Protocols," *Redes de Computadores*, vol. 2, pp. 454-473, 1978.
- TAN, K., SONG, J., ZHANG, Q., e SRIDHARN, M.:** "A Compound TCP Approach for

- Redes de alta velocidade e longa distância," *Proc. INFOCOM Conf.*, IEEE, pp. 1-12, 2006.
- TANENBAUM, AS:** *Modern Operating Systems*, 3^a ed., Upper Saddle River, NJ: Prentice Hall, 2007.
- TANENBAUM, AS, e VAN STEEN, M. :** *Sistemas Distribuídos: Princípios e Paradigmas*, Upper Saddle River, NJ: Prentice Hall, 2007.
- TOMLINSON, RS:** " Selecionando Números de Sequência," *Proc. SIGCOMM / SIGOPS Interprocess Commun. Workshop*, ACM, pp. 11-23, 1975.

Página 924

900

LISTA DE LEITURA E BIBLIOGRAFIA

INDIVÍDUO. 9

- TUCHMAN, W. :** " Hellman Presents No Shortcut Solutions to DES," *IEEE Spectrum*, vol. 16, pp. 40-41, julho de 1979.
- TURNER, JS:** " Novos rumos nas comunicações (ou qual o caminho para as informações Idade)," *IEEE Commun. Revista*, vol. 24, pp. 8-15, outubro de 1986.
- UNGERBOECK, G. :** " Modulação Codificada em Treliça com Conjuntos de Sinais Redundantes Parte I: Introdução-produção," *IEEE Commun. Revista*, vol. 25, pp. 5-11, fevereiro de 1987.
- VALADE, J. :** *PHP & MySQL for Dummies*, 5^a ed., Nova York: John Wiley & Sons, 2009.
- VARGHESE, G. :** *Network Algorithmics*, San Francisco: Morgan Kaufmann, 2004.
- VARGHESE, G., and LAUCK, T. :** " Hashed and Hierarchical Timing Wheels: Data Structures for the Efficient Implementation of a Timer Facility," *Proc. 11th Symp. em Operating Systems Prin.*, ACM, pp. 25-38, 1987.
- VERIZON BUSINESS:** *Relatório de investigações de violação de dados de 2009*, Verizon, 2009.
- VITERBI, A. :** *CDMA: Principles of Spread Spectrum Communication*, Englewood Cliffs, NJ: Prentice Hall, 1995.
- VON AHN, L., BLUM, B., e LANGFORD, J. :** " Telling Humans and Computers Apart Automaticamente," *Commun. do ACM*, vol. 47, pp. 56-60, fevereiro de 2004.
- WAITZMAN, D., PARTRIDGE, C., e DEERING, S. :** " Distance Vector Multicast Routing Protocolo," *RFC 1075*, novembro de 1988.
- WALDMAN, M., RUBIN, AD e CRANOR, LF:** " Publius: A Robust, Inviolável, Censorship-Resistant Web Publishing System," *Proc. Nono USENIX Security Symp.*, USENIX, pp. 59-72, 2000.
- WANG, Z., and CROWCROFT, J. :** " SEAL Detects Cell Misordering," *Rede IEEE Revista*, vol. 6, pp. 8-9, julho de 1992.
- WANT, R. :** *RFID Explained*, San Rafael, CA: Morgan Claypool, 2006.
- WARNEKE, B., LAST, M., LIEBOWITZ, B., e PISTER, KSJ:** " Smart Dust: Comunicando com um Computador Milímetro Cúbico," *IEEE Computer*, vol. 34, pp. 44-51, janeiro 2001.
- WAYNER, P. :** *Disappearing Cryptography: Information Hiding, Steganography, and Watermarking*, 3rd ed., San Francisco: Morgan Kaufmann, 2008.
- WEI, D., CHENG, J., LOW, S., e HEGDE, S. :** " FAST TCP: Motivation, Architecture, Algoritmos, desempenho," *IEEE / ACM Trans. on Networking*, vol. 14, pp. 1246-1259, Dezembro de 2006.
- WEISER, M. :** " The Computer for the Twenty-First Century," *Scientific American*, vol. 265, pp. 94-104, setembro de 1991.
- WELBOURNE, E., BATTLE, L., COLE, G., GOULD, K., RECTOR, K., RAYMER, S., BALAZINSKA, M., e BORRIELLO, G. :** " Construindo a Internet das Coisas Usando RFID," *IEEE Internet Computing*, vol. 13, pp. 48-55, maio de 2009.
- WITTENBURG, N. :** *Understanding Voice Over IP Technology*, Clifton Park, NY: Delmar Cengage Learning, 2009.

Página 925

SEC. 9,2

BIBLIOGRAFIA ALFABÉTICA

901

- WOLMAN, A., VOELKER, G., SHARMA, N., CARDWELL, N., KARLIN, A., e LEVY, H. :** " Na escala e desempenho do cache de proxy da Web cooperativo," *Proc. Dia 17 Symp. em sistemas operacionais Prin.*, ACM, pp. 16-31, 1999.
- WOOD, L., IVANCIC, W., EDDY, W., STEWART, D., NORTHAM, J., JACKSON, C., e DA SILVA CURIEL, A. :** " Uso do Delay-Tolerant Networking Bundle Protocol do espaço," *Proc. 59º Congresso Internacional de Astronáutica*, Federação Internacional de Astronáutica, pp. 3123-3133, 2008.

- WU, T.** : " Network Neutrality, Broadband Discrimination, " *Journal on Telecom. e Alta tecnologia. Lei* , vol. 2, pp. 141-179, 2003.
- WYLIE, J., BIGRIGG, MW, STRUNK, JD, GANGER, GR, KILICCOTE, H., e KHOSLA, PK:** " Survivable Information Storage Systems, " *IEEE Computer* , vol. 33, pp. 61-68, agosto de 2000.
- YU, T., HARTMAN, S., e RAEBURN, K.** : " The Perils of Unauthenticated Encryption: Kerberos Versão 4, " *Proc. Simpósio NDSS* , Internet Society, fevereiro de 2004.
- YUVAL, G.** : " How to Swindle Rabin, " *Cryptologia* , vol. 3, pp. 187-190, julho de 1979.
- ZACKS, M.** : " Antiterrorist Legislation Expands Electronic Snooping, " *IEEE Internet Computing* , vol. 5, pp. 8-9, novembro – dezembro. 2001.
- ZHANG, Y., BRESLAU, L., PAXSON, V., e SHENKER, S.** : " Sobre as Características e Origens of Internet Flow Rates, " *Proc. SIGCOMM 2002 Conf.* , ACM, pp. 309-322, 2002
- ZHAO, B., LING, H., STRIBLING, J., RHEA, S., JOSEPH, A., e KUBIATOWICZ, J.** : " Tapestry: A Resilient Global-Scale Overlay for Service Deployment, " *IEEE J. on Áreas Selecionadas em Comun.* , vol. 22, pp. 41-53, janeiro de 2004.
- ZIMMERMANN, PR:** *The Official PGP User Guide* , Cambridge, MA: MIT Press, 1995a.
- ZIMMERMANN, PR:** *PGP: Source Code and Internals* , Cambridge, MA: MIT Press, 1995b.
- ZIPF, GK:** *Comportamento Humano e o Princípio do Menor Esforço: Uma Introdução ao Humano Ecology* , Boston: Addison-Wesley, 1949.
- ZIV, J., and LEMPEL, Z.** : " A Universal Algorithm for Sequential Data Compression, " *IEEE Trans. on Information Theory* , vol. IT – 3, pp. 337–343, maio de 1977.

Página 926

Esta página foi intencionalmente deixada em branco

Página 927

ÍNDICE

Página 928

Esta página foi intencionalmente deixada em branco

Página 929

ÍNDICE

Números

- 1-CSMA persistente, [266](#)
- 3GPP (ver Terceira Geração Projeto de Parceria)
- Codificação 4B / 5B, [128](#) , 292
- Codificação 8B / 10B, [129](#), 295
- Ethernet de 10 Gigabit, [296](#) -297
- Codificação 64B / 66B, [297](#)
- Ethernet 100Base-FX, [292](#)
- Ethernet 100Base-T4, [291](#) -292
- 802.11 (consulte IEEE 802.11)
- Ethernet 1000Base-T, [295](#) -296

UMA

- Lei A, [153](#) , 700
- AAL5 (consulte ATM Adaptation Layer 5)
- Notação de sintaxe abstrata [1](#) , 809
- Ponto de acesso, [19](#), 70, 299
- camada de transporte, [509](#)
- Datagrama reconhecido, [37](#)
- Reconhecimento cumulativo, [238](#), 558
- duplicado, [577](#)
- seletivo, [560](#), 580
- Relógio de confirmação, TCP, [574](#)

Quadro de confirmação, [43](#)
ACL (*consulte* link sem conexão assíncrona)
Página do servidor ativo, [676](#)
ActiveX, [858](#) -859
Controle ActiveX, [678](#)
Rede ad hoc, [70](#), 299, 389-392
roteamento, [389](#) -392
Vetor de distância sob demanda ad hoc, [389](#)
Adaptação, taxa, [301](#)
Salto de frequência adaptável, Bluetooth, [324](#)
Algoritmo de roteamento adaptativo, [364](#)
Adaptive árvore caminhada protocolo, [275](#) -277
ADC (*consulte* Conversor Analógico para Digital)
Lei de diminuição multiplicativa de aumento aditivo, [537](#)
Protocolo de resolução de endereço, [467](#) -469
gratuito, [469](#)
Proxy de protocolo de resolução de endereço, [469](#)
Endereçando, [34](#)
IP classful, [449](#) -451
camada de transporte, [509-512](#)
[905](#)

Página 930

906

ÍNDICE

Roteador adjacente, [478](#)
Controle de admissão, [395](#), 397-398, 415-418
ADSL (*consulte* Assinante Digital Assimétrico Linha)
Codificação de áudio avançada, [702](#)
Advanced Encryption Standard, [312](#), 783-787
Advanced Mobile Phone System, [65](#), 167-170
Agência de Projetos de Pesquisa Avançada, [56](#)
Codificação de vídeo avançada, [710](#)
AES (*consulte* o padrão de criptografia avançado)
Agregação, rota, [447](#)
AH (*consulte* o cabeçalho de autenticação)
AIFS (*consulte* Arbitration InterFrame Space)
AIMD (*ver* Aditivo Aumento Multiplicativo
Lei de redução)
Interface aérea, [66](#), 171
AJAX (*consulte* JavaScript assíncrono e XML)
Akamai, [745](#) -746
Algoritmo
roteamento adaptativo, [364](#)
aprendizagem retroativa, [333](#), 335
Bellman-Ford, [370](#)
backoff exponencial binário, [285](#) -286
controle de congestionamento, [392](#) -404
Dijkstra's, [369](#)
codificação, [550](#)
encaminhamento, [27](#)
criptografia de dados internacionais, [842](#)
Karn's, [571](#)
balde furado, [397](#), 407-411
prefixo correspondente mais longo, [448](#)
loteria, [112](#)
Nagle's, [566](#)
roteamento da camada de rede, [362-392](#)
não adaptativa, [363](#) -364
encaminhamento de caminho reverso, [381](#), 419
Rivest Shamir Adleman, [794](#) -796
roteamento, [27](#), 362-392
balde de tokens, [408](#) -411
Alias, [617](#), 619, 630
Alocação, canal, [258](#) -261
ALOHA, [72](#)
puro, [262](#) -264
com fenda, [264](#) -266

Inversão de marca alternativa, [129](#)
AMI (*ver* Inversão de Marca Alternativa)
Modulação de mudança de amplitude, [130](#)
AMPS (*consulte* Sistema Avançado de Telefonia Móvel)
Conversor analógico para digital, [699](#)
Andreessen, Marc, [646](#) -647
Anomalia, taxa, [309](#)
Repostador anônimo, [861](#) -863
ANSNET, [60](#)
Antena, setorizada, [178](#)
Antheil, George, [108](#)
Roteamento anycast, [385](#) - 386
AODV (*consulte* Vetor de distância sob demanda ad-hoc roteamento)
AP (*ver* Ponto de Acesso)
Apocalipse dos dois elefantes, [51](#) -52
Applet, [678](#)
Inscrição
negócios, [3](#)
Web, [4](#)
Camada de aplicação, [45](#) , 47-48
rede de entrega de conteúdo, [734](#)-757
tabela de hash distribuída, [753](#) -757
Sistema de Nome de Domínio, [611](#) -623
e-mail, [623](#) -646
multimídia, [607](#) -734
World Wide Web, [646](#) -697
Gateway de nível de aplicativo, [819](#)
APSD (*consulte* Entrega automática de economia de energia)
Espaço interframe de arbitragem, [308](#)
Visão geral da arquitetura, Web, [647](#) -649
Arquitetura e serviços, e-mail, [624](#)-626
Área, sistema autônomo
backbone, [476](#)
stub, [477](#)
Roteador de borda de área, [476](#)
ARP (*consulte* Protocolo de Resolução de Endereço)
ARPA (*veja* Agência de Projetos de Pesquisa Avançada)
ARPANET, [55](#) - 59
ARQ (*ver* solicitação de repetição automática)
AS (*ver* Sistema Autônomo)
ASK (*ver* Amplitude Shift Keying)
ASN.1 (*consulte* a notação de sintaxe abstrata 1)
ASP (*consulte* Active Server Pages)
Proporção, vídeo, [705](#)
Associação, IEEE 802.11, [311](#)
Encaminhamento garantido, [423](#) -424
Linha de assinante digital assimétrica, [94](#) , 124, 147, 248-250
vs. cabo, [185](#)
Link assíncrono sem conexão, [325](#)
E / S assíncrona, [682](#)
Javascript e XML assíncronos, [679](#) -683
Modo de transferência assíncrona, [249](#)
AT&T, 55, [110](#)
ATM (*consulte* Modo de transferência assíncrona)
Camada de adaptação ATM [5](#), 250

reflexão, [829](#)
repetição, [836](#)
Atenuação, [102](#), 109
Atributo
certificado criptográfico, [808](#)
HTML, [664](#)
Leilão, espectro, [112](#)
Áudio
digital, [699](#) -704
streaming, [697](#) -704
Compressão de áudio, [701](#) -704
Autenticação, [35](#)
IEEE 802.11, [311](#)
Needham-Schroeder, [836](#) -838
usando o centro de distribuição de chaves, [835](#)-838
Cabeçalho de autenticação, [815](#) -816
Protocolo de autenticação, [827](#) -841
Autenticação usando um segredo compartilhado, [828](#) - 833
Autenticação usando Kerberos, [838](#) -840
Autenticação usando chaves públicas, [840](#) -841
Authenticode, [858](#)
Registro [oficial](#), [620](#)
Autocorrelação, [176](#)
Autonegociação, [293](#)
Entrega automática de economia de energia, [307](#)
Solicitação de repetição automática, [225](#), 522
Sistema autônomo, [432](#), 437, 472-476, 474
Autoresponder, [629](#)
AVC (*consulte* Codificação de Vídeo Avançada)

B

Frame B, [712](#)
Backbone, Internet, [63](#)
Área de backbone, [476](#)
Roteador de backbone, [476](#)
Contrapressão, salto a salto, [400](#) -401
Backscatter, RFID, [74](#), 329
Algoritmo de aprendizagem retroativa, [333](#), 335
Ponte de aprendizagem retroativa, [335](#) -336
Sinal balanceado, [129](#) -130
Largura de banda, [91](#)
Alocação de banda, [531](#) -535
Eficiência de largura de banda, [126](#) -127
Produto de atraso de largura de banda, [233](#), 267, 597
Baran, Paul, [55](#)
Sequência de Barker, [302](#)
Estação base, [19](#), 70
Controlador de estação base, [171](#)
Ethernet Base-T, [295](#) -296
Codificação Base64, [634](#)
Sinal de banda base, [91](#), 130
Transmissão de banda base, [125](#) -130
Método básico de bitmap, [270](#)
Taxa Baud, [127](#), 146
Protocolo BB84, [773](#)
Quadro de farol, [307](#)
Concurso de beleza, para alocação de espectro, [112](#)
Bell, Alexander Graham, [139](#)
Bell Operating Company, [142](#)
Algoritmo de roteamento Bellman-Ford, [370](#)
Transponder de tubo curvo, [116](#)
Soquete Berkeley, [500](#) -507
Serviço de melhor esforço, [318](#) - 319
BGP (*consulte* Protocolo de Gateway de Borda)
Computador big-endian, [439](#)
Protocolo de contagem binária, [272](#) -273
Algoritmo de backoff exponencial binário, [285](#) -286
Chaveamento de mudança de fase binária, [130](#)
Codificação bipolar, [129](#)
Ataque de aniversário, [804](#) -806
Taxa de bits, [127](#)
Recheio de bits, [199](#)

Protocolo de mapa de bits, [270](#) -271
BitTorrent, [750](#) –753
par sufocado, [752](#)
pedaço, [751](#)
free-rider, [752](#)
leecher, [752](#)
semeadora, [751](#)
enxame, [751](#)
estratégia tit-for-tat, [752](#)
torrent, [750](#)
rastreador, [751](#)
par não chocado, [752](#)
Blaatand, Harald, [320](#)
Cifra de bloco, [779](#)
Código de bloco, [204](#)

Página 932

908

ÍNDICE

Bluetooth, [18](#) , 320-327
salto de frequência adaptativo, [324](#)
aplicações, [321](#) -322
arquitetura [320](#) -321
estrutura do quadro, [325](#) -327
link, [324](#)
camada de ligação, [324](#) -325
emparelhamento, [320](#)
piconet, [320](#)
perfil, [321](#)
pilha de protocolo, [322](#) -323
camada de rádio, [324](#)
scatternet, [320](#)
emparelhamento seguro, [325](#)
segurança [826](#) -827
Bluetooth SIG, [321](#)
BOC (*consulte* Bell Operating Company)
Corpo, tag HTML, [625](#)
Protocolo de gateway de fronteira, [432](#) , 479-484
Botnet, [16](#) , 628
Roteador de limite, [477](#)
BPSK (*consulte* Binary Phase Shift Keying)
Ponte, [332](#) –342
aprendizagem retroativa, [335](#) -336
em comparação com outros dispositivos, [340](#) –342
aprendizagem, [334](#) -337
árvore geradora, [337](#) - 340
uso, [332](#) -333
Banda larga, [63](#) , 147-151
Banda larga sem fio, [312](#) –320
Canal de controle de transmissão, [173](#)
Rede de transmissão, [17](#)
Roteamento de transmissão, [380](#) -382
Tempestade de transmissão, [344](#), 583
Transmitindo, [17](#), 283
Navegador, [648](#)
extensão, [859](#) -860
aplicativo auxiliar, [654](#)
plug-in, [653](#) –654, 859
BSC (*consulte* o controlador da estação base)
Balde, vazando, [397](#)
Ataque da brigada de balde, [835](#)
Buffer, [222](#), 238, 290, 341
Pacote, rede tolerante a atrasos, [601](#)
Protocolo de pacote, [603](#) -605
Tráfego intenso, [407](#)
Bush, Vannevar, [647](#)
Aplicação de negócios, [3](#)
Fluxo de bytes, confiável, [502](#)
Preenchimento de bytes, [198](#)

C

CA (*consulte Autoridade de Certificação*)
Cabeçalho do cabo, [23](#), 63, 179
Internet a cabo, [180](#) -182
Modem a cabo, [63](#), 183–185, 183–195
Sistema de terminação de modem a cabo, [63](#), 183
Televisão a cabo, [179](#)-186
Cache
ARP [468](#)- 469
DNS, [620](#)-[622](#), 848-850
envenenado, [849](#)
Web, [656](#), 690-692
Cifra de César, [769](#)
Gerenciamento de chamadas, [169](#)
Acoplamento capacitivo, [129](#)
Capacidade, canal, [94](#)
CAPTCHA, [16](#)
Endereço [comercial](#), [387](#)
Carnívoro, [15](#)
Extensão da operadora, Ethernet, [294](#)
Acesso múltiplo de detecção de operadora, [72](#), 266-269
1-persistente, [266](#)
detecção de colisão, [268](#) -269
não persistente, [267](#)
p-persistente, [267](#)
Detecção de portadora, [260](#)
Ethernet de nível de operadora, [299](#)
Folha de estilo em cascata, [670](#)-672
Fiação de categoria 3, [96](#)
Fiação de categoria 5, [96](#)
Fiação de categoria 6, [96](#)
Fiação de categoria 7, [96](#)
CCITT (*consulte União Internacional de Telecomunicações*)
CCK (*veja Codificação Complementar*)
CD (*ver Rascunho do Comitê*)
CDM (*consulte Multiplexação por divisão de código*)
CDMA (*consulte Acesso Múltiplo por Divisão de Código*)
CDMA2000, [175](#)
CDN (*consulte Rede de distribuição de conteúdo*)
Celular, telefone móvel, [167](#), 249
Telefone celular, [165](#)
primeira geração, [166](#)-170
segunda geração, [170](#) -174
terceira geração, [65](#) -69, 174-179
Estação base celular, [66](#)
Rede celular, [65](#)
Certificado
criptográfico, [807](#) -809
X. [509](#), 809-810

Página 933

ÍNDICE

909

Lista de revogação de certificado, [813](#)
Autoridade de certificação, [807](#)
Caminho de certificação, [812](#)
CGI (*consulte Interface de gateway comum*)
Cadeia de confiança, [812](#)
Protocolo de desafio-resposta, [828](#)
Canal
concessão de acesso, [174](#)
controle de transmissão, [173](#)
controle comum, [174](#)
controle dedicado, [174](#)
apagamento, [203](#)
multiacesso, [257](#)
paging, [174](#)
acesso aleatório, [174](#), 257
Alocação de canal, [258](#) -261

dinâmico, [260](#) -261
Capacidade do canal, [94](#)
Sinalização associada ao canal, [155](#)
Soma de verificação, [211](#)
CRC, [210](#)
Fletcher, [212](#)
Seqüência de chips, CDMA, [136](#)
Pacote de choke, [399](#) –400
Ponto bloqueado, BitTorrent, [752](#)
Acorde, [754](#) -757
mesa de dedo, [756](#)
chave, [754](#)
Ataque de texto simples escolhido, [769](#)
Dispersão cromática, [103](#)
Crominância, vídeo, [706](#)
Chunk, BitTorrent, [751](#)
CIDR (*consulte* Classless InterDomain Routing)
Rede de distribuição de Cintent, [743](#) -748
Cifra, [766](#)
AES, [783](#) -787
César, [769](#)
substituição monoalfabética, [770](#)
Rijndael, [784](#) -787
substituição, [767](#)-770
de chave simétrica, [778](#) -787
transposição, [771](#) -772
Modo de encadeamento de bloco de criptografia , [788](#) –789
Modo de feedback de cifra, [789](#) - 790
Modos de criptografia , [787](#) –792
Texto cifrado, [767](#)
Ataque somente texto cifrado, [769](#)
Circuito, [35](#)
virtual, [249](#)
Comutação de circuito, [161](#) -162
Clark, David, [51](#), 81
Rede Classe A, [450](#)
Rede classe B, [450](#)
Rede classe C, [450](#)
Roteamento baseado em classe, [421](#)
Endereçamento classful, IP, [449](#) - 451
Ethernet clássica, [21](#) , 280, 281-288
Roteamento entre domínios sem classe, [447](#) - 449
Liberado para enviar, [279](#)
Fraude de cliques, [697](#)
Cliente, [4](#)
Lado do cliente na Web, [649](#) –652
Lado do cliente dinâmica de geração da página Web, [676](#) -678
Lado do cliente na Web, [649](#) –652
Stub do cliente, [544](#)
Modelo cliente-servidor, [4](#)
Chip Clipper, [861](#)
Recuperação de relógio, [127](#) -129
Computação em nuvem, [672](#)
CMTS (*consulte* Sistema de terminação de modem a cabo)
Cabo coaxial, [97](#) -98
Código, criptográfico, [766](#)
Acesso múltiplo por divisão de código, [66](#) , 108, 135, 170
Multiplexação por divisão de código, [135](#) -138
Taxa de código, [204](#)
Assinatura de código, [858](#)
Codec, [153](#)
Palavra-código, [204](#)
Colisão, [260](#)
Detecção de colisão, CSMA, [268](#) -269
Domínio de colisão, [289](#)
Protocolo livre de colisão, [269](#)-273
Pentear, artefato visual, [705](#)
Projeto do Comitê, [79](#)
Canal de controle comum, [174](#)
Interface de gateway comum, [674](#)
Sinalização de canal comum, [155](#)

Meio de comunicação, [5](#)
Satélite de comunicação, [116](#) -125
Segurança de comunicação, [813](#) -827
Sub-rede de comunicação, [24](#)
Antena comunitária de televisão, [179](#) -180
Companding, [154](#)
Comparação do OSI e TCP / IP
modelos, [49](#) -51
Codificação de código complementar, [302](#)
Compressão
áudio, [701](#) -704
cabecalho, [593](#) -595
vídeo, [706](#) -712

Página 934

910

ÍNDICE

Computador, vestível, [13](#)
Rede de computadores (*ver* Rede)
GET condicional, HTTP, [691](#)
Confidencialidade, [35](#)
Congestionamento, rede, [35](#) , 392-404
Prevenção de congestionamento, [398](#)
Colapso do congestionamento, [393](#)
TCP, [572](#)
Controle de congestão
convergência, [534](#) -535
camada de rede, [392](#) -404
provisionamento, [395](#)
TCP, [571](#) -581
Janela de congestionamento, TCP, [571](#)
Conexão, HTTP, [684](#) -686
Estabelecimento de conexão, [512](#) -517
TCP, [560](#) -562
Gerenciamento de conexão, TCP, [562](#) -565
Liberação de conexão, [517](#) -522
TCP, [562](#)
Reutilização de conexão, HTTP, [684](#)
Serviço, orientado a conexão [35](#) -38, 359-361
implementação, [359](#) -361
Serviço sem conexão, [35](#) -38, 358-359
implementação, [358](#) -359
Conectividade, [6](#), 11
Constelação, [146](#)
Diagrama de constelação, [131](#)
Comprimento da restrição, [207](#)
Contato, rede tolerante a atrasos, [601](#)
Conteúdo e tráfego da Internet, 7360738
Rede de distribuição de conteúdo, [743](#) -748
Distribuição de conteúdo, [734](#) -757
Transformação de conteúdo, [694](#)
Sistema de contenção, [262](#)
Mídia contínua, [699](#)
Canal de controle, transmissão, [173](#)
Lei de controle, [536](#)
Convergência, [372](#)
controle de congestionamento [534](#)-535
Código convolucionar, [207](#)
Cookie, [15](#)
SYN, [561](#)
Web, [658](#) -662
Copyright, [867](#) -869
Telefone sem fio, [165](#)
Rede central, [66](#)
Árvore baseada em núcleo, [384](#)
Problema de contagem até o infinito [372](#)-373
Modo contador, [791](#) -792
Bater recuperação, [527](#) -530
CRC (*ver* verificação de redundância cíclica)

Crítica da OSI e TCP / IP, [51](#) -53
CRL (*ver* lista de revogação de certificado)
Correlação cruzada, [176](#)
Criptoanálise, [768](#), 792-793
diferencial, [792](#) -793
linear, [793](#)
Certificado criptográfico, [807](#) - 809
Chave criptográfica, [767](#)
Princípios de criptografia, [776](#) -778
Rodada criptográfica, [780](#)
Criptografia, [766](#) -797
AES, [312](#)
certificado, [807](#) -809
texto cifrado, [767](#)
DES, [780](#) -784
Princípio de Kerckhoff, [768](#)
chave, [767](#)
one-time pad, [772](#) -773
P-box, [779](#)
texto simples, [767](#)
-chave pública, [793](#) -797
quantum, [773](#) -776
Rijndael, [312](#)
S-box, [779](#)
segurança pela obscuridade, [768](#)
de chave simétrica, [778](#) -793
DES triplo, [782](#) -783
vs. código, [766](#)
fator de trabalho, [768](#)
Criptologia, [768](#)
CSMA (*consulte* Carrier Sense Multiple Access)
CSMA com prevenção de colisão, [303](#)
CSMA com detecção de colisão, [268](#)
CSMA / CA (*ver* CSMA com colisão
Evasão)
CSMA / CD (*ver* CSMA com Colisão
Detecção)
CSNET, [59](#)
CSS (*consulte* Cascading Style Sheet)
CTS (*consulte* Limpar para enviar)
CubeSat, [123](#)
Reconhecimento cumulativo, [238](#), 558
TCP, [568](#)
Transferência de custódia, rede tolerante a atrasos, [604](#)
Comutação direta, [36](#) , 336
Cybersquatting, [614](#)
Verificação de redundância cíclica, [212](#)
Repostador Cypherpunk, [862](#)

ÍNDICE

911

D

D-AMPS (*consulte* Digital Advanced Mobile Sistema Telefônico)
DAC (*consulte* Conversor digital para analógico)
Daemen, Joan, [784](#)
Daemon, [554](#)
DAG (*ver* Gráfico Acíclico Direcionado)
Data center, [64](#)
Serviço de entrega de dados, IEEE 802.11, [312](#)
O padrão de encriptação de dados, [780](#) -784
Quadro de dados, [43](#)
Camada de enlace de dados, [43](#) , 193-251
recheio de bits, [199](#)
enchimento de bytes, [197](#)
problemas de design, [194](#) -215
protocolos elementares, [215](#) -244
protocolos de exemplo, [244](#) -250

protocolos de janela deslizante, [226](#) -244
protocolo parar e esperar, [222](#) -223
Troca de camada de enlace de dados, [332](#) -349
Protocolo de enlace de dados, [215](#) -250
ADSL, [248](#) -250
elementar, [215](#) -244
exemplos, [215](#) -250
pacote sobre SONET, [245](#) -248
janela deslizante, [226](#) -244
pare e espere, [222](#) -223
Dados sobre especificação de interface de serviço a cabo, [183](#)
Datagrama, [37](#), 358
Protocolo de controle de congestionamento de datagrama, [503](#)
Rede de datagrama, [358](#)
Serviço de datagrama, comparação com VCs, [361](#) -362
Davies, Donald, [56](#)
DB (*ver* decibel)
DCCP (*ver* Datagrama Congestionamento Controlado Protocolo)
DCF (*ver* Função de Coordenação Distribuída)
Espaçamento entre quadros DCF, [308](#)
DCT (*ver* Transformação Discreta de Cosseno)
Ataque DDoS (*consulte* Negação de serviço distribuída ataque)
Padrão de fato, [76](#)
Padrão de jure, [76](#)
Decibel, [94](#), 699
Decodificação, áudio, [701](#)
Canal de controle dedicado, [174](#)
Deep Web, [696](#)
Gateway padrão, [469](#)
Zona livre de padrão, [446](#)
Round robin de déficit, [414](#)
Atraso, fila, [164](#)
Rede tolerante a atrasos, [599](#) -605
arquitetura, [600](#) -603
transferência de custódia, [604](#)
protocolo, [603](#) -605
Confirmação atrasada, TCP, [566](#)
Zona desmilitarizada, [819](#)
Ataque de negação de serviço, [820](#)
distribuído, [821](#) -822
Multiplexação por divisão de comprimento de onda densa, [160](#)
DES (*consulte* o padrão de criptografia de dados)
Problemas de design
camada de enlace de dados, [194](#)-202
redes rápidas, [586](#) -590
rede, [33](#) -35
camada de rede, [355](#) -362
camada de transporte, [507](#)-530
Roteador designado, [375](#), 478
Compartilhamento de área de trabalho, [5](#)
Porta de destino, [453](#) -454
Driver de dispositivo, [215](#)
DHCP (*consulte* Protocolo de configuração de host dinâmico)
DHT (*consulte* a Tabela de Hash Distribuída)
Base diagonal, em criptografia quântica, [774](#)
Modem dial-up, [62](#)
Controle de diálogo, [44](#)
Criptoanálise diferencial, [792](#) -793
Atendimento diferenciado, [421](#) -424, 440, 458
Protocolo de Diffie-Hellman, [833](#) -835
DIFS (*consulte* DCF InterFrame Spacing)
Sistema digital avançado de telefonia móvel, [170](#)
Áudio digital, [699](#) -704
Digital Millennium Copyright Act, [14](#), 868
Modulação digital, [125](#)
Assinatura digital, [797](#)-806
Padrão de assinatura digital, [800](#)
Linha de assinante digital, [62](#), 147-151
Multiplexador de acesso à linha de assinante digital, [62](#), 150

Vídeo digital, [704](#) -712
Conversor digital para analógico, [700](#)
Digitalizando sinais de voz, [153](#)-154
Digram, [770](#)
Dijkstra, Edsger, [367](#)
Algoritmo de Dijkstra, [369](#)
Gráfico acíclico direto, [365](#)
Espectro de propagação de sequência direta, [108](#)
Gráfico acíclico direcionado, [365](#)
Diretiva, HTML, [663](#)
Diretório, PKI, [812](#)

Página 936

912

ÍNDICE

DIS (*ver* Rascunho da Norma Internacional)
Desassociação, IEEE 802.11, [311](#)
Descoberta, caminho MTU, [556](#)
Transformação discreta de cosseno, MPEG, [707](#)
Multitom discreto, [148](#)
Dispersão, cromática, [103](#)
Disposição, mensagem, [628](#)
Rede tolerante a interrupções, [600](#)
Protocolo de roteamento multicast de vetor de distância, [383](#)
Roteamento do vetor de distância, [370](#) -378
Distorção, [700](#)
Função de coordenação distribuída, [304](#)
Ataque distribuído de negação de serviço, [821](#) -822
Tabela de hash distribuída, [753](#) -757
Sistema distribuído, [2](#)
Serviço de distribuição, IEEE 802.11, [311](#)
Sistema de distribuição, [299](#)
Padrão Ethernet DIX, [281](#), 283
DMCA (*consulte* Digital Millennium Copyright Act)
Aviso de remoção DMCA, [14](#)
DMT (*ver* Discrete MultiTone)
DMZ (*consulte* Zona Desmilitarizada)
DNS (*consulte* Sistema de Nome de Domínio)
DNS Nome Espaço, [612](#) -616
Spoofing DNS, [848](#) -850
DNSsec (*consulte* Segurança do Sistema de Nome de Domínio)
DOCSIS (*consulte* Data Over Cable Service
Especificação de interface)
Modelo de objeto de documento, [679](#)
DOM (*consulte* Document Object Model)
Domínio
colisão, [289](#)
frequência, [133](#)
Sistema de nomes de domínio, [59](#), 611-623
registro oficial, [620](#) -622
cybersquatting, [614](#)
registro de recurso de domínio, [616](#) -619
servidor de nomes, [619](#) –623
espaço de nome, [613](#)
registrador, [613](#)
registro de recurso, [616](#) -619
pesquisa reversa, [617](#)
falsificação, [851](#)
domínio de nível superior, [613](#)
zona, [851](#) -852
Ataque DoS (*ver* ataque de negação de serviço)
Era ponto com, [647](#)
Notação decimal pontilhada, [443](#)
Proxy downstream, [742](#)
Rascunho de Padrão Internacional, [79](#)
Rascunho da norma, [82](#)
DSL (*consulte* Digital Subscriber Line)
DSLAM (*consulte* Digital Subscriber Line Access Multiplexer)

DTN (*consulte* Rede Tolerante a Delay)
Confirmação duplicada, TCP, [577](#)
DVMRP (*ver* Distance Vector Multicast
Protocolo de Roteamento)
DWDM (*ver* Divisão Densa de Comprimento de Onda
Multiplexing)
Tempo de permanência, [324](#)
Alocação dinâmica de canais, [260](#) -261
Seleção de frequência dinâmica, [312](#)
Protocolo de configuração de host dinâmico, [470](#)
HTML dinâmico, [676](#)
Página dinâmica, Web, [649](#)
Roteamento dinâmico, [364](#)
Página da Web dinâmica, [649](#), 672-673
Geração dinâmica de página da web
lado do cliente, [676](#)-678
do lado do servidor, [673](#) -676

E

Comércio eletrônico, [6](#), 9
E-mail (*ver* E-mail)
Linha E1, [155](#)
EAP (*consulte* Protocolo de autenticação extensível)
Saída antecipada, [484](#)
BCE (*ver* modo Livro de Código Eletrônico)
ECMP (*consulte* Equal Cost MultiPath)
ECN (*consulte* Notificação Explícita de Congestionamento)
EDE (*consulte* o modo Criptografar e Descriptografar)
EDGE (*consulte* Taxas de dados aprimoradas para evolução GSM)
EEE (*consulte* o modo Encrypt Encrypt Encrypt)
EIFS (*consulte* Extended InterFrame Spacing)
Eisenhower, Dwight, [56](#)
Espectro eletromagnético, [105](#) –109, 111–114
Modo de livro de código eletrônico, [787](#) -788
Comércio eletrônico, [6](#)
Correio eletrônico (*ver* Email)
Código de produto eletrônico, [327](#)
Fluxo de elefante, [737](#)
Email, [5](#), 623-646
arquitetura e serviços, [624](#)-626
registro [oficial](#), [620](#)
codificação base64, [634](#)
corpo, [625](#)
registro em cache, [620](#)

Página 937

ÍNDICE

913

Email (*continuação*)
envelope, [625](#)
entrega final, [643](#)
IMAP, [644](#) -645
servidor de correio, [624](#)
envio de correio, [641](#)
caixa de correio, [625](#)
formato de mensagem, [630](#)
transferência de mensagem, [624](#) , 637-643, 642
MIME, [633](#)
resolução de nome, [620](#)
retransmissão de correio aberto, [642](#)
POP3, [644](#)
codificação para impressão entre aspas, [634](#)
bloco de assinatura, [629](#)
protocolo de transferência de correio simples, [625](#)
agente de transferência, [624](#) -625
agente do usuário, [624](#) , 626
agente de férias, [629](#)
Webmail, [645](#) –646
X400, [629](#)
Cabeçalho de email, [625](#)

Leitor de e-mail, [626](#)
Segurança de e-mail, [841](#) -846
Emoticon, [623](#)
Carga útil de segurança de encapsulamento, [817](#)
Encapsulamento, pacote, [387](#)
Codificação, 4B / 5B, [292](#)
áudio, [701](#) -704
vídeo, [706](#) -712
Ethernet 4B / 5B, [292](#)
Ethernet 8B / 10B, [295](#)
Ethernet 64B / 66B, [297](#)
Criptografar, descriptografar, criptografar modo, [782](#)
Criptografar criptografar modo criptografar, [782](#)
Criptografia, link, [765](#)
Escritório final, [140](#)
Argumento ponta a ponta, [357](#), 523
Ponto final, multiplexação, [527](#)
Taxas de dados aprimoradas para evolução GSM, [178](#)
Rede corporativa, [19](#)
Entidade, transporte, [496](#)
Envelope, [625](#)
EPC (ver Código de Produto Eletrônico)
EPON (ver Ethernet PON)
Multipath de custo igual, [476](#)
Apagamento, [716](#)
Canal de apagamento, [203](#)
Controle de erro, [200](#)–201
camada de transporte, 522-527
Correção de erros, [34](#)
Detecção de erro, [33](#)
Síndrome do erro, [207](#)
Código de correção de erros, [204](#) –209
Código de detecção de erro, [209](#) -215
ESMTP (consulte SMTP estendido)
ESP (veja Encapsulando Segurança
Carga útil)
Serviço de eternidade, [864](#)
Ethernet, [20](#), 280–299
10 gigabit, [296](#) -297
100Base-FX, [292](#)
100Base-T4, [292](#)
1000Base-T, [295](#) -296
Base-T, [295](#) -296
grau de operadora, [299](#)
clássico, [21](#), 281-288
DIX, [281](#), 283
rápido, [290](#) -293
gigabit, [293](#) -296
Subcamada MAC, [280](#) -299
modo promíscuo, [290](#)
retrospectiva, 298-299
comutado, [20](#), 288-290
Extensão da operadora Ethernet, [294](#)
Codificação Ethernet
4B / 5B, [292](#)
64B / 66B, [297](#)
8B / 10B, [295](#)
Bursting de frame Ethernet, [294](#)
Cabeçalho Ethernet, [282](#)
Hub Ethernet, [288](#)
Jumbo frame Ethernet, [296](#)
Desempenho Ethernet, [286](#) -288
Ethernet PON, [151](#) -152
Porta Ethernet, [20](#)
Switch Ethernet, [20](#), 289
EuroDOCSIS, [183](#)
EWMA (ver Movimento Exponencialmente Ponderado
Média)
Forwarding acelerada, [422](#) -423
Notificação explícita de congestionamento, [400](#)
Decadência exponencial, [738](#)

Média móvel exponencialmente ponderada, [399](#), 570
Problema do terminal exposto, [278](#) -280
Linguagem de marcação de hipertexto estendida, [681](#)
Espaçamento estendido entre quadros, [309](#)
SMTP estendido, [639](#)
Superquadro estendido, [154](#)
Protocolo de autenticação extensível, [824](#)

Página 938

914

ÍNDICE

Linguagem de marcação extensível, [680](#)
Transformação de linguagem de folha de estilo extensível, [681](#)
Cabeçalho de extensão, IPv6, [461](#) -463
Protocolo de gateway externo, [431](#) , 474

F

Facebook, [8](#)
Fila justa, [412](#)
Doutrina de uso justo, [869](#)
Fast Ethernet, [290](#) -293
Rede rápida, design, [586](#) -590
Recuperação rápida, TCP, [578](#)
Retransmissão rápida, TCP, [577](#)
Processamento rápido de segmento, [590](#) -593
FCFS (*ver* pacote First-Come First-Served
agendamento)
FDD (*consulte* Duplex de Divisão de Freqüência)
FDDI (*consulte* Interface de dados distribuída de fibra)
FDM (*consulte* Multiplexação por divisão de frequência)
FEC (*consulte* Classe de Equivalência de Encaminhamento)
FEC (*ver* Correção de Erro Forward)
FEC (*consulte* Classe de Equivalência de Encaminhamento)
Interface de dados distribuída de fibra, [272](#)
Nó de fibra, [180](#)
Fibra óptica, [99](#) -105
em comparação com o fio de cobre, [104](#) -105
Fibra para casa, [63](#) , 100, 151
Canal de fibra, [298](#)
Campo, vídeo, [705](#)
FIFO (*ver* programação de pacotes First-In First-Out)
Protocolo de transferência de arquivos, [455](#) , 623
Filtragem, entrada, [487](#)
Entrega final, email, [643](#)
Mesa de dedo, acorde, [756](#)
Firewall, [818](#) -821
stateful, [819](#)
Programação de pacotes por ordem de chegada, [412](#)
Primeira geração de rede de telefonia móvel, [166](#) -170
Programação de pacotes primeiro a entrar, primeiro a sair, [412](#)
Sem fio fixo, [11](#)
Byte de sinalização, [198](#)
Multidão Flash, [747](#) , 748
Soma de verificação de Fletcher, [212](#)
Algoritmo de inundação, [368](#) - 370
Controle de fluxo, [35](#) , 201-202
camada de transporte, 522-527
Especificação de fluxo, [416](#)
Pegada, satélite, [119](#)
Região proibida, [514](#) -515
Agente estrangeiro, [388](#) , 487
Forma, Web, [667](#) -670
Correção de erro antecipada, [203](#) , 715
Encaminhamento, [363](#)
assegurada, [423](#) -424
acelerado, [422](#) -423
Algoritmo de encaminhamento, [27](#)
Classe de equivalência de encaminhamento, [472](#)
Análise de Fourier, [90](#)
Série de Fourier, [90](#)

Transformada de Fourier, [135](#) , 702
Fragmento
quadro, [307](#)
pacote, [433](#)
Fragmentação, pacote, [432](#) -436
Quadro, [194](#)
reconhecimento, [43](#)
farol, [307](#)
dados, [43](#)
Frame bursting, Ethernet, [294](#)
Fragmento de quadro, [307](#)
Cabeçalho do quadro, [218](#)
Quadro estrutural
Bluetooth, [325](#) -327
IEEE 802.11, [309](#) –310
IEEE 802.16, [319](#) –320
Enquadramento, [197](#) -201
Free-rider, BitTorrent, [752](#)
Óptica de espaço livre, [114](#)
Liberdade de expressão, [863](#) -865
Frequência, espectro eletromagnético, [106](#)
Duplex de divisão de frequência, [169](#) , 317
Multiplexação por divisão de frequência, [132](#) -135
Salto de frequência, Bluetooth, [324](#)
Espectro de propagação de salto de frequência, [107](#)
Mascaramento de frequência, psicoacústica, [703](#)
Reutilização de frequência, [65](#)
Seleção de frequência, dinâmica, [312](#)
Chaveamento de mudança de frequência, [130](#)
Novidade das mensagens, [778](#)
Front end, servidor Web, [739](#)
FSK (*consulte* Frequency Shift Keying)
FTP (*veja* Programa de Transferência de Arquivos)
FttH (*consulte* Fiber to the Home)
Link full-duplex, [97](#)
Futuro do TCP, [581](#) -582
Fuzzball, [59](#)

Página 939

ÍNDICE
915
G
G.711 padrão, [728](#)
G.dmt, [149](#)
G.lite, [150](#)
Gatekeeper, multimídia, [728](#)
Gateway, [28](#)
nível de aplicação, [819](#)
padrão, [469](#)
mídia, [68](#)
multimídia, [728](#)
Nó de suporte GPRS do gateway, [68](#)
Gateway mobile switching center, [68](#)
Gen 2 RFID, [327](#) –331
Serviço geral de pacote de rádio, [68](#)
Polinômio gerador, [213](#)
GEO (*ver* Órbita Terrestre Geoestacionária)
Órbita terrestre geoestacionária, [117](#)
Satélite geoestacionário, [117](#)
GGSN (*consulte* Gateway GPRS Support Node)
Gigabit Ethernet, [293](#) –296
PON com capacidade de gigabit, [151](#)
Sistema de Posicionamento Global, [12](#) , 121
Sistema global para comunicação móvel, [65](#),
[170](#)– 174
Globalstar, [122](#)
Gmail, [15](#)
GMSC (*consulte* Gateway Mobile Switching Center)
Protocolo Go-back-n, [232](#) -239

Goodput, [393](#), 531
GPON (*consulte* PON capaz de Gigabit)
GPRS (*consulte* Serviço de rádio de pacote geral)
GPS (*ver* Sistema de Posicionamento Global)
ARP gratuita, [469](#), 487
Gray, Eliseu, [139](#)
Código Gray, [132](#) Grupo, 153
Grupo, hierarquia de telefone, [153](#)
GSM (*consulte* Sistema Global para Celular comunicação)
Banda de guarda, [133](#)
Tempo de guarda, [135](#)
Mídia de transmissão guiada, [85](#) –105, 95–105

H

H.225 padrão, [729](#)
H.245 padrão, [729](#)
H.264 padrão, [710](#)
H.323
em comparação com o protocolo SIP, [733](#) -734
padrão, [728](#) -731
Link half-duplex, [97](#)
Código de Hamming, [206](#) –207
Distância de Hamming, [205](#)
Handoff, [68](#) -69, 168
difícil, [178](#)
macio, [178](#)
Transferência (*ver* transferência)
Decodificação de decisão difícil, [208](#)
Harmônico, [90](#)
Código de autenticação de mensagem hash, [817](#)
HDLC (*consulte* Controle de link de dados de alto nível)
HDTV (*consulte* TeleVision de alta definição)
Headend, cabo, [63](#) , 179
Cabeçalho, [625](#)
email, [625](#)
Ethernet, [282](#)
IPv4, [439](#) –442
IPv6, [458](#) –463
Extensão IPv6, [461](#) –463
pacote, [31](#)
Segmento TCP, [557](#) -560
Compressão de cabeçalho, [593](#) –595
robusto, [594](#)
Previsão de cabeçalho, [592](#)
Aplicativo auxiliar, navegador, [654](#)
Hertz, [106](#)
Hertz, Heinrich, [106](#)
HF RFID, [74](#)
HFC (*consulte* Hybrid Fiber Coax)
Problema de terminal oculto, [278](#)
Roteamento hierárquico, 378-380
Televisão de alta definição, [705](#)
Controle de link de dados de alto nível, [199](#) , 246
Marca d'água alta, [719](#)
História da Internet, 54-61
HLR (*ver* Registro de localização de casa)
HMAC (*consulte* Código de autenticação de mensagem hash)
Agente doméstico, [387](#), 486
Localização residencial, [386](#)
Registro de localização residencial, [171](#)
Rede doméstica, [6](#) 10
Servidor de assinante residencial, [69](#)
Contrapressão salto a salto, [400](#) -401
Host, [23](#)
celular, [386](#)
Design de host para redes rápidas, 586-590
Hospedagem, [64](#)

- Roteamento de batata quente, [484](#)
Hotspot, [11](#)
HSS (*consulte* Servidor de assinante doméstico)
HTML (*consulte* a linguagem de marcação de hipertexto)
HTTP (*consulte* Protocolo de transferência de hipertexto)
HTTPS (*consulte* Secure HyperText Transfer Protocol)
Hub, [340](#) -342
em comparação com a ponte e comutador, [340](#) -342
Ethernet, [288](#)
satélite, [119](#)
Fibra híbrida coaxial, [180](#)
Hyperlink, [648](#)
Hipertexto, [647](#)
Hypertext markup language, [663](#) -670
atributo, [664](#)
diretiva, [663](#)
tag, [663](#) -666
Protocolo de transferência de hipertexto, [45](#) , 649, 651, 683-693
obtenção condicional, [691](#)
conexão, [684](#) -686
reutilização de conexão, [684](#)
método, [686](#) -688
conexão paralela, [686](#)
conexão persistente, [684](#)
esquema, [650](#)
seguro [853](#)
Hz (*ver* Hertz)
- Eu**
- IAB (*ver* Internet Activities Board)
ICANN (*ver* Internet Corporation for Assigned Nomes e Número)
ICMP (*consulte* Internet Control Message Protocol)
IDEA (*ver* International Data Encryption Algoritmo)
IEEE 802.11, [19](#) , 299-312
arquitetura, [299](#) -301
associação, [311](#)
autenticação, [311](#)
comparação com IEEE 802.16, [313](#) -314
serviço de entrega de dados, [312](#)
dissociação, [311](#)
serviço de distribuição, [311](#)
estrutura de quadro, [309](#) -310
serviço de integração, [312](#)
Subcamada MAC, [299](#) -312
Protocolo de subcamada MAC, [303](#) - 309
IEEE 802.11 (*continuação*)
camada física, [301](#) -303
serviço de privacidade, [312](#)
pilha protocolo, [299](#) -301
reassociação, [311](#)
segurança, [823](#) -826
serviços, [311](#) -312
transmitir controle de potência, [312](#)
IEEE 802.11a, [302](#)
IEEE 802.11b, [17](#) , 301–302
IEEE 802.11g, [203](#)
IEEE 802.11i, [823](#)
IEEE 802.11n, [302](#) -303
IEEE 802.16, [179](#) , 313-320
arquitetura, [314](#) -315
comparação com IEEE 802.11, [313](#) -314
estrutura de quadro, [319](#) -320
Protocolo de subcamada MAC, [317](#) -319
camada física, [316](#) -317
pilha de protocolo, [314](#) -315
variando, [317](#)
IEEE 802.1Q, [346](#) -349

IEEE 802.1X, [824](#)
IEEE (*consulte* Instituto de Elétrica e Eletrônica Engenheiros)
IETF (*consulte* Internet Engineering Task Force)
IGMP (*consulte* Internet Group Management Protocol)
IKE (*consulte* Internet Key Exchange)
IMAP (*ver* Internet Message Access Protocol)
IMP (*ver* Interface Message Processor)
Sistema de telefonia móvel aprimorado, [166](#)
IMT-2000, [174](#)–[175](#)
IMTS (*consulte* Sistema de telefonia móvel aprimorado)
Sinalização em banda, [155](#)
Bandas industriais, científicas, médicas, [70](#), 112
Inetd, [554](#)
Infrared Data Association, [114](#)
Transmissão infravermelha, [114](#)
Filtragem de entrada, [487](#)
Protocolo de conexão inicial, [511](#)
Vetor de inicialização, [788](#)
Formulário de entrada, Web, [667](#)–[670](#)
Mensagens instantâneas, [8](#)
Instituto de Engenheiros Elétricos e Eletrônicos, [79](#)
Serviços integrados, 418–421
Serviço de integração, IEEE 802.11, [312](#)
Propriedade intelectual, [867](#)
Protocolo interdomínio, [431](#)
Roteamento entre domínios, [474](#)
Operadora Interexchange, [143](#)

Página 941

ÍNDICE
917
Interface, [30](#)
ar, [66](#), 171
Processador de mensagem de interface, [56](#)–[57](#)
Protocolo de gateway interior, [431](#), 474
Entrelaçamento, vídeo, [705](#)
Intercalação, [716](#)
Roteador interno, [476](#)
Algoritmo de criptografia de dados internacional, [842](#)
International Mobile Telecommunication-2000,
[174](#)–[175](#)
Padrão Internacional, [78](#)–[79](#)
Padrão Internacional IS-95, [170](#)
Organização Internacional de Padrões, [78](#)
União Internacional de Telecomunicações, [77](#)
Internet, [2](#), 28
Arquitectura, [61](#)–[64](#)
backbone, [63](#)
cabو, [180](#)–[182](#)
protocolos de controle, [465](#)–[470](#)
daemon, [554](#)
história, [54](#)–[61](#)
interplanetário, [18](#)
troca de chave, [815](#)
protocolo de acesso à mensagem, [644](#)–[645](#)
multicast, [484](#)–[488](#)
protocolo versão 4, [439](#)–[455](#)
protocolo versão 6, [455](#)–[465](#)
rádio, [721](#)
Camada TCP / IP, [46](#)–[47](#)
Quadro de Atividades da Internet, [81](#)
Conselho de Arquitetura da Internet, [81](#)
Protocolo de mensagem de controle da Internet, [47](#)
Corporação da Internet para nomes atribuídos
e números, [444](#), 612
Força-Tarefa de Engenharia da Internet, [81](#)
Ponto de troca de Internet, [63](#), 480
Protocolo de gerenciamento de grupo da Internet, [485](#)

Internet por cabo, [180](#) -182
Protocolo da Internet (IP), [47](#), 438-488
CIDR, [447](#) -449
endereçamento classful, [449](#) -451
controle, [465](#) -470
mensagem de controle, [47](#)
protocolos de controle, [465](#) -470
notação decimal com pontos, [443](#)
gestão de grupo, [485](#)
Endereços IP, [442](#) -455
acesso à mensagem, [644](#) -645
celular, [485](#) -488
sub-rede, [444](#) -446
Protocolo de Internet (*continuação*)
versão 4, [439](#) -442
versão 5, [439](#)
versão 6, [455](#) -465
controvérsias da versão 6, 463-465
cabeçalhos de extensão da versão 6, [461](#) -463
versão 6 do cabeçalho principal, [458](#) -461
Protocolo de Internet versão 4, [439](#) -455
Protocolo de Internet versão 6, [455](#) -465
Internet Research Task Force, [81](#)
Provedor de serviços de Internet, [26](#), 62
Sociedade da Internet, [81](#)
Padrão da Internet, [82](#)
Telefonia pela Internet, [698](#), 725
Internetwork, [25](#), 28-29, 424-436
Roteamento Internetwork, [431](#) -432
Internetworking, [34](#), 424-436
camada de rede, [19](#), 424-436
Camada de rede de internetworking, [424](#)-436
Tronco entre escritórios, [141](#)
Internet Interplanetária, [18](#)
Protocolo intradomínio, [431](#)
Roteamento intradomínio, [474](#)
Intranet, [64](#)
Intruso, segurança, [767](#)
Multiplexação inversa, [527](#)
IP (*ver* protocolo da Internet)
Endereço IP, [442](#) -455
CIDR, [447](#) -449
classful, [449](#) -451
NAT, [451](#) -455
prefixo, [443](#) -444
Segurança IP, [814](#) -818
modo de transporte, [815](#)
modo túnel, [815](#)
Telefonia IP, [5](#)
Televisão IP, [9](#), 721
IPsec (*consulte* segurança IP)
IPTV (*ver* IP TeleVision)
IPv4 (*consulte* Protocolo de Internet, versão 4)
IPv5 (*consulte* Protocolo de Internet, versão 5)
IPv6 (*consulte* Protocolo de Internet, versão 6)
IrDA (*consulte* Infrared Data Association)
Iridium, [121](#)
IRTF (*consulte* Internet Research Task Force)
IS (*ver* Padrão Internacional)
IS-95, [170](#)
IS-IS, [378](#), 474
ISAKMP (*consulte* Internet Security Association e
Protocolo de Gerenciamento de Chaves)

ISP (*consulte* Provedor de Serviços de Internet)
Rede ISP, [26](#)
Consulta iterativa, [622](#)
ITU (*ver* International Telecommunication Union)
IV (*ver* vetor de inicialização)
IXC (*consulte* IntereXchange Carrier)
IXP (*consulte* Internet eXchange Point)

J

Segurança de miniaplicativo Java, [857](#) -858
Máquina virtual Java, [678](#)
Java Virtual Machine, [857](#)
JavaScript, [676](#), 859
Página Javaserver, [675](#)
Jitter, [406](#), 698
Controle de jitter, [550](#) -552
Grupo conjunto de especialistas em fotografia, [706](#)
JPEG (*consulte* Joint Photographic Experts Group)
Padrão JPEG, [706](#) -709
JSP (*consulte* a página JavaServer)
Jumbo frame, Ethernet, [296](#)
Jumbograma, [462](#)
JVM (*consulte* Java Virtual Machine)

K

Algoritmo de Karn, [571](#)
KDC (*consulte* Centro de distribuição de chaves)
Timer de Keepalive, TCP, [571](#)
Lei de Kepler, [116](#)
Kerberos, [838](#) -840
reino, [840](#)
Princípio de Kerckhoff, [768](#)
Chave
Chord, [754](#)
criptográfico, [767](#)
Centro de distribuição de chaves, [807](#), 828
autenticação usando, [835](#) -836
Caução de chave, [861](#)
Keystream, [790](#)
Ataque de reutilização de fluxo de chaves, [791](#)
Ataque de texto simples conhecido, [769](#)

eu

L2CAP (*ver* Adaptação de Controle de Link Lógico Protocolo)
Roteador de borda de etiqueta, [472](#)
Roteador comutado por etiqueta, [472](#)
Troca de etiqueta, [360](#), 470-474
Lamarr, Hedy, [107](#) -108
LAN (*consulte* Rede Local)
LAN, virtual, [21](#)
LATA (*consulte* Acesso Local e Área de Transporte)
Camada
aplicação, [45](#), 47-48, 611-758
link de dados, [193](#) -251
IEEE 802.11 física, [301](#) -303
IEEE 802.16 física, [316](#) -317
rede, [29](#), 355-489
físico, [89](#) -187
sessão, [44](#) -45
transporte, [44](#), 495-606
LCP (*consulte* Protocolo de Controle de Link)
LDPC (*ver* verificação de paridade de baixa densidade)
Algoritmo de balde com vazamento, [397](#), 407-411
Ponte aprendizagem, [334](#) -337
LEC (*consulte* Operadora de troca local)
Leecher, BitTorrent, [752](#)
LEO (*ver* satélite de órbita terrestre baixa)
LER (*consulte* Label Edge Router)
Nível, rede, [29](#)
Transmissão de luz, [114](#) -116
Protocolo de contenção limitada, [274](#) -277
Código de linha, [126](#)

Código linear, [204](#)
Criptoanálise linear, [793](#)
Ligaçāo
sem conexāo assíncrona, [325](#)
Bluetooth, [324](#)
full-duplex, [97](#)
half-duplex, [97](#)
orientado para conexāo síncrona, [325](#)
Protocolo de controle de link, [245](#)
Criptografia de link, [765](#)
Camada de link
Bluetooth, [324](#)–325
TCP / IP, [46](#)
Roteamento de estado de link, [373](#)–378
Computador Little-endian, [429](#)
LLC (*consulte* Controle de link lógico)
Balanceamento de carga, servidor Web, [740](#)
Descarte de carga, [395](#), 401–403

Página 943

ÍNDICE

919

Política de redução de carga
leite, [401](#)
vinho, [401](#)
Acesso local e área de transporte, [142](#)
Rede local, [19](#)
virtual, [342](#)–349
Escritório central local, [21](#)
Operadora de câmbio local, [142](#)
Loop local, [140](#), 144–152
Portabilidade do número local, [144](#)
Controle de link lógico, [283](#), 310
Protocolo de adaptação de controle de link lógico, [323](#)
Rede longa e gorda, [595](#)–599
Evolução de longo prazo, [69](#), 179, 314
Algoritmo de prefixo de correspondência mais longa, [448](#)
Compressão de áudio sem perdas, [701](#)
Compressão de áudio com perdas, [701](#)
Algoritmo de loteria, [112](#)
Verificação de paridade de baixa densidade, [209](#)
Satélite em órbita terrestre baixa, [121](#), 121–123
Marca d'água baixa, [718](#)
LSR (*consulte* Roteador Comutado por Etiqueta)
LTE (*ver* evolução de longo prazo)
Luminância, vídeo, [706](#)

M

M-commerce, [12](#)
MAC (*consulte* Controle de Acesso Médio)
Protocolo subcamada MAC, [303](#)–309, 317–319
IEEE 802.11, [303](#)–309
IEEE 802.16, [317](#)–319
MACA (*consulte* Acesso múltiplo com colisão
Evasão)
Macroblock, MPEG, [711](#)
Mídia magnética, [95](#)–96
MAHO (*consulte* Mobile Assisted HandOff)
Servidor de correio, [624](#)
Envio por correio, [624](#), 637, 641
Caixa de correio, [625](#)
Lista de mala direta, [625](#)
Manutenção, rota, [391](#)–392
MAN (*ver* Rede de Área Metropolitana)
Ataque man-in-the-middle, [835](#)
Codificação Manchester, [127](#)
MANET (*ver* Rede ad hoc móvel)
Linguagem de marcação, [663](#), 680
Parâmetros de empacotamento, [544](#)
Max-min equidade, [532](#)–534

Tamanho máximo do segmento, TCP, [559](#)
Unidade máxima de transferência, [433](#), 556
Caminho máximo da unidade de transmissão, [433](#)
Maxwell, James Clerk, [105](#)
MCI, [110](#)
MD5, [804](#)- 807
Medições de desempenho da rede, [584](#) -586
Gateway de mídia, [68](#)
Subcamada de controle de acesso médio, [43](#) , 257-350,
[320](#) -327
Bluetooth, [320](#) -327
banda larga sem fio, [312](#) -320
alocação de canal, [258](#) -261
Ethernet, [280](#) -299
IEEE 802.11, [299](#) -312
protocolos de acesso múltiplo, [261](#)-280
LANs sem fio, [299](#) - 312
Satélite em órbita terrestre média, [121](#)
MEO (*ver* Satélite de Órbita Terrestre Média)
Merkle, Ralph, [796](#)
Resumo da mensagem, [800](#) a 801
Disposição da mensagem, [628](#)
Formato de mensagem, [630](#)
Cabeçalho da mensagem, [688](#) - 690
Verificação de integridade da mensagem, [825](#)
Troca de mensagens, [600](#)
Transferência de mensagem, [637](#) -643, 642
Agente de transferência de mensagens, [624](#)
Metafile, [714](#)
Metcalfe, Robert, [6](#)
Método, HTTP, [686](#) -688
Unidades métricas, [82](#) -83
Rede de área metropolitana, [23](#)
MFJ (*ver* Julgamento Final Modificado)
MGW (*consulte* Media GateWay)
MIC (*consulte* Verificação de integridade da mensagem)
Experiência de Michelson-Morley, [281](#)
Microcélula, [168](#)
Transmissão de microondas, [110](#) -114
Middlebox, [740](#)
Middleware, [2](#)
Leite, política de redução de carga, [401](#)
MIME (*consulte* Multipurpose Internet Mail Extension)
Tipo MIME, [652](#) -655
MIMO (*consulte* Multiple Input Multiple Output)
Minislot, [184](#)
Espelhando um site, [744](#)
Rede ad hoc móvel, [389](#) - 392
Transferência assistida móvel, [174](#)

Página 944

920

ÍNDICE

Segurança de código móvel, [857](#) –860
Comércio móvel, [12](#)
Host móvel, [386](#)
roteamento, [386](#) -389
Protocolo de Internet móvel, [485](#) –488
Protocolo IP móvel, [485](#) -488
Telefone celular, [164](#) -179
Sistema de telefonia móvel
primeira geração, [166](#)-170
segunda geração, [170](#) -174
terceira geração, [65](#) -69, 174-179
Centro de comutação móvel, [68](#) , 168
Telefone móvel, [164](#) -179
Central de comutação de telefonia móvel, [168](#)
Sistema de telefonia móvel, [164](#) -179
Usuário móvel, [10](#) a 13

Mobile Web, [693](#) -695
Sem fio móvel, [11](#)
Mockapetris, Paul, [52](#)
Modem, [145](#)
cabô, [63](#) , 183-195
dial-up, [62](#)
V.32, [146](#)
V.34, [146](#)
V.90, [147](#)
Julgamento final modificado, [142](#)
Modulação, [130](#) -132
chaveamento de mudança de amplitude, [130](#) -131
BPSK, [302](#)
digital, [125](#)
multitom discreto, [149](#)
chaveamento de mudança de frequência, [130](#) -131
chaveamento de mudança de fase, [130](#) -131
código de pulso, [153](#)
chaveamento de mudança de fase em quadratura, [131](#)
trelíça codificada, [146](#)
Cifra de substituição monoalfabética, [770](#)
MOSPF (*consulte* Multicast OSPF)
Grupo de especialistas em cinema, [709](#)
Mouse, [737](#)
Fluxo do mouse, [737](#)
MP3, [702](#)
MP4, [702](#)
MPEG (*consulte* Motion Picture Experts Group)
Compressão MPEG, [709](#) - 712
tipos de quadro, [712](#)
MPEG-1, [710](#)
MPEG-2, [710](#)
MPEG-4, [710](#)
Padrões MPEG, [709](#) -710
MPLS (*consulte* MultiProtocol Label Switching)
MSC (*consulte* Mobile Switching Center)
MSS (*consulte* o tamanho máximo do segmento)
MTSO (*consulte* Central de comutação de telefonia móvel)
MTU (*ver* Unidade Máxima de Transferência)
Lei Mu, [700](#)
Mu-law, [153](#)
Canal multiacesso, [257](#)
Rede multiacesso, [475](#)
OSPF multicast, [383](#)
Roteamento multicast, [382](#) , 382-385
Multicast, [17](#) , 283, 382
Internet, [484](#) -488
Roteamento multidestino, [380](#)
Multihoming, [481](#)
Rede multihop, [75](#)
Multimídia, [697](#) - 734, 699
Telefonia via Internet, [728](#) -734
controle de jitter, [550](#) -552
transmissão ao vivo, [721](#)-724
MP3, [702](#) -704
RTSP, [715](#)
streaming de áudio, [699](#) -704
video on demand, [713](#) -720
videoconferência, [725](#) -728
voz sobre IP, [728](#) -734
Fibra multimodo, [101](#)
Desvanecimento multipercorso, [70](#) , 111
Protocolo de acesso múltiplo, [261](#) -280
Acesso múltiplo com prevenção de colisão, [279](#)
Múltiplas entradas múltiplas saídas, [303](#)
Multiplexação, [125](#) , 152-160
ponto final, [527](#)
inverso, [527](#)
estatístico, [34](#)
Trocada de rótulo multiprotocolo, [357](#) , 360, 471-474
Roteador multiprotocolo, [429](#)

Extensão de correio da Internet multiuso, 632–637
Servidor da Web multithread, 656
Multitom, discreto, 148
Mutlimedia, streaming de vídeo, 704 – 712

N

Algoritmo de Nagle, 566
Resolução de nome, 620
Servidor de nomes, 619 –623
Nomenclatura (veja Endereçamento)

Página 945

ÍNDICE

921

Nomeação, seguro, 848 –853
NAP (consulte Ponto de acesso à rede)
NAT (consulte Tradução de endereço de rede)
Caixa NAT, 453
Instituto Nacional de Padrões e Tecnologia, 79 , 783
Agência de Segurança Nacional, 782
NAV (consulte Vetor de alocação de rede)
NCP (consulte Protocolo de Controle de Rede)
Comunicação de campo próximo, 12
Autenticação Needham-Schroeder, 836 –838
Protocolo de negociação, 35
Rede
ad hoc, 70 , 299, 389-392
ALOHA, 72
transmissão, 17
celular, 65
tolerante a retardo, 599 -605
empresa, 19
telefone celular de primeira geração, 166 -170
lar, 6-10
área local, 19
área metropolitana, 23
multiacesso, 475
multihop, 75
óptica passiva, 151
ponto a ponto, 7
desempenho, 582 -599
área pessoal, 18
ponto a ponto, 17
linha de força, 10 , 22
telefone público comutado, 68 , 139
escalável, 34
telefone celular de segunda geração, 170 -174
sensor, 13 , 73-75
social, 8
stub, 481
de terceira geração de telefonia móvel, 65 -69, 174-179
utilizações, 3 -16
circuito virtual, 358
virtual privado, 4 , 26
área ampla, 23
Acelerador de rede, 215
Ponto de acesso à rede, 60 -61
Tradução de endereço de rede, 452 –455
Vetor de alocação de rede, 305
Arquitetura de rede, 31
Protocolo de controle de rede, 245
Questões de design de rede, 33 -35
Hardware de rede, 17 -29
Placa de interface de rede, 202 , 215
Dispositivo de interface de rede, 149
Camada de rede, 43 -44, 355-489
controle de congestionamento, 392 -404
problemas de design, 355 -362
Internet, 436 -488

internetworking, [424](#) -436
qualidade de serviço, [404](#) -424
algoritmos de roteamento, [362](#) -392
Neutralidade da rede, [14](#)
Sobreposição de rede, [430](#)
Medição de desempenho de rede, [584](#) -586
Protocolo de rede (*ver* protocolo)
Ponto de acesso de serviço de rede, [509](#)
Provedor de serviço de rede, [26](#)
Software de rede, [29](#) -41
Padronização de rede, [75](#) -82
NFC (*consulte* Near Field Communication)
NIC (*consulte* Placa de Interface de Rede)
NID (*consulte* Dispositivo de interface de rede)
NIST (*ver* Instituto Nacional de Padrões e
Tecnologia)
Nó B, [66](#)
Identificador de nó, [754](#)
Codificação invertida sem retorno para zero, [127](#)
Modulação sem retorno a zero, [125](#)
Algoritmo não adaptativo, [363](#) - 364
Nonce, [824](#)
CSMA não persistente, [267](#)
Cookie da Web não persistente, [659](#)
Não-repúdio, [798](#)
Notificação, congestionamento explícito, [400](#)
NRZ (*ver* modulação sem retorno a zero)
NRZI (*consulte* a codificação sem retorno para zero invertido)
NSA (*ver* Agência de Segurança Nacional)
NSAP (*consulte* Ponto de acesso de serviço de rede)
NSFNET, [59](#) -61
Frequência de Nyquist, [94](#), 146, 153

O

OFDM (*ver* Divisão de Frequência Ortogonal
Multiplexing)
OFDMA (*ver* Divisão de Frequência Ortogonal
Acesso múltiplo)
One-time pad, [772](#) -773
Roteamento de cebola, [863](#)
Retransmissão de correio aberto, [642](#)

Página 946

922

ÍNDICE

Abra o caminho mais curto primeiro, [378](#)
Abra o caminho mais curto primeiro roteamento, [474](#) -479
Interconexão de Sistemas Abertos, [41](#) -45
comparação com TCP / IP, [49](#) -51
Interconexão de sistemas abertos,
camada de aplicação, [45](#)
crítica, [51](#) -53
camada de enlace de dados, [43](#)
camada de rede, [43](#) -44
camada física, [43](#)
camada de apresentação, [45](#)
modelo de referência, [41](#) -45
camada de sessão, [44](#) -45
camada de transporte, [44](#)
Princípio de otimização, [364](#) -365
Identificador único organizacional, [283](#)
Acesso múltiplo por divisão de frequência ortogonal, [316](#)
Multiplexação por divisão de frequência ortogonal, [71](#),
[133](#)-[134](#), 302
Sequência ortogonal, [136](#)
OSI (*consulte* Interconexão de sistemas abertos)
OSPF (*consulte* Roteamento do caminho mais curto aberto primeiro)
OSPF (*consulte* Roteamento do caminho mais curto aberto primeiro)
OUI (*consulte* Identificador Organizacionalmente Único)
Sinalização fora de banda, [155](#)

Sobreposição, [430](#)

rede, [430](#)

Superprovisionamento, [404](#)

P

P-box, criptográfico, [779](#)

Frame P, [611](#) - 712

P-persistente CSMA, [267](#)

P2P (*ver* rede ponto a ponto)

Pacote, [17](#), 36

Encapsulamento de pacotes, [387](#)

Filtro de pacotes, [819](#)

Fragmentação de pacote, [432](#) –436, 433

Cabeçalho do pacote, [31](#)

Pacote sobre SONET, [245](#) –248

Programação de pacotes, [411](#) –414

Troca de pacotes, [162](#) –164

armazenar e encaminhar, [356](#)

Trem de pacotes, [736](#)

Página, Web, [647](#)

Canal de paging, [174](#)

Emparelhamento, Bluetooth, [320](#)

PAN (*ver* Rede de Área Pessoal)

PAR (*ver* reconhecimento positivo com

Protocolo de retransmissão)

Conexão paralela, HTTP, [686](#)

Parâmetros, empacotamento, [544](#)

Bit de paridade, [210](#)

Pacote de paridade, [716](#)

Passband, [91](#)

Sinal de banda passante, [130](#)

Transmissão de banda passante, [125](#), 130-132

Rede óptica passiva, [151](#)

RFID passivo, [73](#)

Senha, [826](#)

Caminho,

sistema autônomo, [481](#)

certificação, [812](#)

unidade de transmissão máxima, [433](#)

Diversidade de caminho, [70](#)

Perda de caminho, [109](#)

Descoberta da unidade de transmissão máxima de caminho, [433](#)

Descoberta de MTU de caminho, [556](#)

Protocolo de vetor de caminho, [481](#)

PAWS (*consulte* Proteção contra embalagem

Números de sequência)

PCF (*ver* Função de Coordenação de Ponto)

PCM (*consulte* Modulação por Código de Pulso)

PCS (*consulte* Personal Communications Service)

Peer, [29](#), 63

Rede ponto a ponto, [7](#), 14, 735, 748-753

BitTorrent, [750](#) –753

distribuição de conteúdo, [750](#) -753

Peering, [481](#)

Comportamento por salto, [421](#)

Codificação perceptual, [702](#)

Desempenho, TCP, [582](#) -599

Problemas de desempenho em redes, [582](#) -599

Medidas de desempenho, rede, [584](#) -586

Perlman, Radia, [339](#)

Temporizador de persistência, TCP, [571](#)

CSMA persistente e não persistente, [266](#) -268

Conexão persistente, HTTP, [684](#)

Cookie persistente, Web, [659](#)

Rede de área pessoal, [18](#)

Serviço de comunicações pessoais, [170](#)

PGP (*consulte* Pretty Good Privacy)

Chaveamento de mudança de fase, [130](#)

Phishing, [16](#)

Telefone (*ver* telefone)

Photon, [774](#) -776, 872

PHP, [674](#)

ÍNDICE

923

Camada física, [89](#) -187
televisão a cabo, [179](#)-186
multiplexação por divisão de código, [135](#) -138
satélites de comunicação, [116](#) -125
fibra ótica, [99](#)-104
multiplexação por divisão de frequência, [132](#) -135
IEEE 802.11, [301](#) -303
IEEE 802.16, [316](#) –317
telemóveis, [164](#) -179
modulação, [125](#) -135
Interconexão de Sistemas Abertos, [43](#)
sistema telefônico, [138](#) -164
multiplexação por divisão de tempo, [135](#)
pares trançados, [96](#) -98
transmissão sem fio, [105](#) -116
Meio físico, [30](#)
Piconet, Bluetooth, [320](#)
Pegando carona, [226](#)
PIM (*consulte* Multicast independente de protocolo)
Ping, [467](#)
Pipelining, [233](#)
Pixel, [704](#)
PKI (*consulte* Infraestrutura de chave pública)
Serviço telefônico simples e antigo, [148](#)
Texto simples, [767](#)
Ponto de reprodução, [551](#)
Plug-in, navegador, [653](#) , 859
Podcast, [721](#)
Poema, árvore abrangente, [339](#)
Função de coordenação de ponto, [304](#)
Ponto de presença, [63](#) , 143
Rede ponto a ponto, [17](#)
Protocolo ponto a ponto, [198](#) , 245
Cache envenenado, [849](#)
Modelo de Poisson, [260](#)
Gerador polinomial, [213](#)
Código polinomial, [212](#)
PON (*consulte* Rede Ótica Passiva)
POP (*ver* Ponto de Presença)
POP3 (*ver* Post Office Protocol)
Porta
destino, [453](#) -454
Ethernet, [20](#)
fonte, [453](#)
TCP, [553](#)
camada de transporte, [509](#)
UDP, [542](#)
Portmapper, [510](#)
Confirmação positiva com retransmissão
protocolo, [225](#)
Administração dos Correios, Telégrafos e Telefone, [77](#)
Protocolo da agência postal, versão 3, [644](#)
POTS (*ver* Plain Old Telephone Service)
Poder, [532](#)
Lei de potência, [737](#)
Rede elétrica, [10](#), 22, 98-99
Modo de economia de energia, [307](#)
Rede elétrica, [10](#), 22
PPP (*ver* protocolo ponto a ponto)
PPP sobre ATM, [250](#)
PPPoA (*consulte* PPP sobre ATM)
Preâmbulo, [200](#)
Predição, cabeçalho, [592](#)
Codificação preditiva, [548](#)
Prefixo, endereço IP, [443](#) -444

Chave pré-mestre, [854](#)
Camada de apresentação, [45](#)
Privacidade muito boa, [842](#)-846
Primitivo, serviço, [38](#)-40
Principal, segurança, [773](#)
Privacidade, [860](#)-861
Amplificação de privacidade, [776](#)
Serviço de privacidade, IEEE 802.11, [312](#)
Porta-chaves privado, [845](#)
Rede privada, virtual, [26](#), 821
Servidor de processo, [511](#)
Protocolo pilha, IEEE 802.11, [299](#)-301
Cifra do produto, [780](#)
Código do produto, eletrônico, [327](#)
Perfil, Bluetooth, [321](#)
Vídeo progressivo, [705](#)
Ethernet de modo promíscuo, [290](#)
Padrão proposto, [81](#)
Proteção contra números de sequência agrupados, [516](#), 560
Protocolo, [29](#)
Janela deslizante de 1 bit, [229](#)-232
caminhada árvore adaptativa, [275](#)-277
resolução de endereço, [467](#)-469
resolução de endereço gratuita, [469](#)
proxy de protocolo de resolução de endereço, [469](#)
protocolos de autenticação, 827-841
BB84, [773](#)
contagem regressiva binária, [272](#)-273
bit-map, [270](#)-271
Bluetooth, [322](#)-323
Pilha de protocolo Bluetooth, [322](#)-323
portal de fronteira, [432](#), 479-484
pacote, [603](#)-605
acesso múltiplo de detecção de operadora, [266](#)-269
desafio-resposta, [828](#)

Página 948

924

ÍNDICE

Protocolo (*continuação*)
livre de colisão, [269](#)-273
CSMA, [266](#)-269
link de dados, [215](#)-250
controle de congestionamento de datagrama, [503](#)
rede tolerante a atrasos, [603](#)-605
Diffie-Hellman, [833](#)-835
roteamento multicast de vetor de distância, [383](#)
notação decimal com pontos Internet, [443](#)
DVMR, [383](#)
configuração de host dinâmica, [470](#)
autenticação extensível, [824](#)
portal externo, [431](#), 474
transferência de arquivo, [455](#), 623
voltar-n, [232](#)-239
transferência de hipertexto, [45](#), 649, 651, 683-693
IEEE 802.11, [299](#)-312
IEEE 802.16, [312](#)-320
conexão inicial, [511](#)
interdomínio, [431](#)
portal interno, [431](#), 474
IP, [438](#)-488
intradomínio, [431](#)
 contenção limitada, [274](#)-277
 controle de link, [245](#)
 adaptação de controle de link lógico, [323](#)
 rede longa e gorda, [595](#)-599
 MAC, [261](#)-280
 IP móvel, [485](#)-488
 acesso múltiplo, [261](#)-280

troca de rótulo multiprotocolo, [360](#), 471-474
roteador multiprotocolo, [429](#)
negociação, [35](#)
rede, [29](#)
controle de rede, [245](#)
pacote sobre SONET, [245](#) –248
vetor de caminho, [481](#)
ponto a ponto, [198](#), 245
POP3, [644](#)
reconhecimento positivo com retransmissão,
[225](#)
streaming em tempo real, [715](#)
tempo real, [606](#)
transporte em tempo real, [546](#)-550
em relação aos serviços, [40](#)
pedido-resposta, [37](#)
reserva, [271](#)
reserva de recursos, [418](#)
repetição seletiva, [239](#) -244
início da sessão, [731](#)-735
Protocolo (*continuação*)
Internet mais simples, [457](#)
transferência de correio simples, [625](#), 638-641
janela deslizante, [226](#) -244, 229–232, 522
SLIP, [245](#)
SOAP, [682](#)
pare e espere, [221](#) -226, 522
fluxo, [503](#), 527
transmissão de controle de fluxo, [503](#), 527
sub-rede da Internet, [444](#) -446
TCP (*consulte* Protocolo de Controle de Transmissão)
integridade de chave temporária, [825](#)
TKIP, [825](#)
passagem de token, [271](#) -272
transporte, [507](#)-530, 541-582
caminhada na árvore, [275](#) -277
UDP, [541](#) -552
utopia, [220](#) -222
aplicativo sem fio, [693](#)
wireless LAN, [277](#) -280
Protocolo 1 (utopia), [220](#) -222
Protocolo 2 (parar e esperar), [221](#) -222
Protocolo 3 (PAR), [222](#) -226
Protocolo 4 (janela deslizante), [229](#) -232
Protocolo 5 (voltar-n), [232](#) -239
Protocolo 6 (repetição seletiva), [239](#) -244
Hierarquia de protocolo, [29](#) -33
Multicast independente de protocolo, [385](#), 485
Camada de protocolo, [34](#)
Pilha de protocolo, [31](#) -32
Bluetooth, [322](#) -323
H.323, [728](#) -731
IEEE 802.11, [299](#) -301
IEEE 802.16, [314](#) -315
OSI, [41](#) -45
TCP / IP, [45](#) -48
Proxy ARP, [469](#)
Cache de proxy, Web, [692](#)
PSK (*consulte* Phase Shift Keying)
PSTN (*consulte* Rede telefônica pública comutada)
Codificação de áudio psicoacústica, [702](#) -703
PTT (*ver* administração de correios, telégrafos e telefones)
Rede telefônica pública comutada, [68](#), 138-164, 139
Autenticação de chave pública usando, [840](#) -841
Criptografia de chave pública, [793](#) -797
outros algoritmos, [796](#) -797
RSA, [794](#) -796
Infraestrutura de chave pública, [810](#)-813
diretório, [812](#)
Anel de chave pública, [845](#)

ÍNDICE

925

Assinatura de chave pública, [799](#) -800

Modulação de código de pulso, [153](#)

ALOHA puro, [262](#) - 265

Sistema push-to-talk, [166](#)

Q

Padrão Q.931, [729](#)

QAM (*consulte* Modulação de amplitude em quadratura)

QoS (*consulte* Qualidade de Serviço)

Programação de tráfego QoS (*consulte* Controle de potência de transmissão)

QPSK (*consulte* Quadrature Phase Shift Keying)

Modulação de amplitude em quadratura, [132](#)

Modificação de mudança de fase em quadratura, [131](#)

Qualidade de serviço, [35](#) , 404-424, 411-414

controle de admissão, [415](#) -418

requisitos de aplicação, [405](#) -406

serviços diferenciados, [421](#)-424

serviços integrados, [418](#)-421

camada de rede, [404](#) -424

requisitos, [405](#) -406

modelagem de tráfego, [407](#) -411

Roteamento de qualidade de serviço, [415](#)

Quantização, MPEG, [707](#)

Ruído de quantização, [700](#)

Criptografia quântica, [773](#) -776

Qubit, [774](#)

Atraso na fila, [164](#)

Teoria de filas, [259](#)

Codificação para impressão entre aspas, [634](#)

R

RA (*ver* Autoridade Regional)

Rede de acesso de rádio, [66](#)

Identificação de radiofrequência, [10](#) , 327-332

ativo, [74](#)

retroespalhamento, [329](#)

geração [2](#) , 327-331

HF, [74](#)

LF, [74](#)

passivo [74](#)

UHF, [73](#) -74

Controlador de rede de rádio, [66](#)

Transmissão de rádio, [109](#) -110

Canal de acesso aleatório, [174](#), 257

Detecção precoce aleatória, [403](#) -404

Variando, [184](#)

IEEE 802.16, [317](#)

RAS (*ver* Registro / Admissão / Status)

Adaptação de taxa, [301](#)

Taxa de anomalia, [309](#)

Regulação de taxa, envio [535](#)-539

Áudio em tempo real, [697](#)

Em tempo real conferêncial, [724](#) -734

Protocolo em tempo real, [606](#)

Protocolo de streaming em tempo real, [715](#)

Protocolos de transporte em tempo real [546](#)-550

Vídeo em tempo real, [697](#)

Reino, Kerberos, [840](#)

Reassociação, IEEE 802.11, [311](#)

Janela de recebimento, [228](#)

Recuperação

relógio, [127](#) -129

acidente, [527](#) -530

rápido, [578](#)

Base retilínea, em criptografia quântica, [774](#)

Consulta recursiva, [621](#)

VERMELHO (*ver* detecção inicial aleatória)

Redundância, em criptografia quântica, [777](#) -778

Código Reed-Solomon, [208](#)
Modelo de referência, [41-54](#),
Interconexão de sistemas abertos, [41](#) -45
TCP / IP, [45](#) -51
Ataque de reflexão, [829](#)
Região, em rede, [379](#)
Autoridade Regional, [811](#)
Registrador, [613](#)
Registro / admissão / status, [729](#)
Relação de protocolos com serviços, [40](#)
Relação de serviços com protocolos, [40](#)
Fluxo de bytes confiável, [502](#)
Remailer
anônimo, [861](#) -863
cyberpunk, [862](#)
Login remoto, [61](#) , 405-406
Chamada de procedimento remoto, [543](#) -546
parâmetros de empacotamento, [544](#)
stubs, [544](#)
Ponto de encontro, [384](#)
Repetidor, [281](#) , 340-342
Ataque de repetição, [836](#)
Pedido de comentários, [81](#)
Solicitar cabeçalho, [688](#)
Pedido de envio, [279](#)
Protocolo de solicitação-resposta, [37](#)

Página 950

926

ÍNDICE

Serviço de solicitação-resposta, [37](#)
Protocolo de reserva, [271](#)
Anel de pacote resiliente, [271](#) , 272
Resolver, [612](#)
Registro de recursos, [616](#)
Conjunto de registros de recursos, [851](#)
Protocolo de reserva de recursos, [418](#)
Compartilhamento de recursos, [3](#)
Cabeçalho de resposta, [688](#)
Retransmissão, rápido, [577](#)
Tempo limite de retransmissão, TCP, [568](#)
Retransmissão temporizador, [570](#) -571
Retrospectiva na Ethernet, [298-299](#)
Pesquisa reversa, [617](#)
Algoritmo de encaminhamento de caminho reverso, [381](#), 419
Certificado de revogação, [812](#) -813
RFC (ver solicitação de comentários)
RFC [768](#), 541
RFC [793](#), 552
RFC [821](#), 625
RFC [822](#), 625, 630, 632, 633, 635, 636, 843, 862
RFC 1058, [373](#)
RFC 1122, [553](#)
RFC 1191, [556](#)
RFC 1323, [516](#), 596
RFC 1521, [634](#)
RFC 1663, [246](#)
RFC 1700, [441](#)
RFC 1939, [644](#)
RFC 1958, [436](#)
RFC 2018, [553](#)
RFC 2109, [659](#)
RFC 2326, [719](#)
RFC 2364, [250](#)
RFC 2410, [814](#)
RFC 2440, [843](#)
RFC 2459, [809](#)
RFC 2535, [851](#), 853
RFC 2581, [553](#)

RFC 2597, [423](#)
RFC 2615, [247](#)
RFC 2616, [683](#), 689
RFC 2854, [635](#)
RFC 2883, [560](#), 580
RFC 2965, [689](#)
RFC 2988, [553](#), 570
RFC 2993, [455](#)
RFC 3003, [635](#)
RFC 3022, [452](#)
RFC 3023, [635](#)
RFC 3119, [717](#)
RFC 3168, [553](#), 558, 581
RFC 3174, [804](#)
RFC 3194, [460](#)
RFC 3246, [422](#)
RFC 3261, [731](#)
RFC 3344, [487](#)
RFC 3376, [485](#)
RFC 3390, [574](#)
RFC 3501, [644](#)
RFC 3517, [580](#)
RFC 3550, [549](#)
RFC 3748, [824](#)
RFC 3775, [488](#)
RFC 3782, [580](#)
RFC 3875, [674](#)
RFC 3963, [488](#)
RFC 3986, [652](#)
RFC 4120, [838](#)
RFC 4306, [815](#)
RFC 4409, [642](#)
RFC 4614, [553](#)
RFC 4632, [447](#)
RFC 4838, [601](#)
RFC 4960, [582](#)
RFC 4987, [562](#)
RFC 5050, [603](#)
RFC 5246, [856](#)
RFC 5280, [809](#)
RFC 5321, [625](#), 633, 639
RFC 5322, [625](#), 630, 631, 632, 633, 760
RFC 5681, [581](#)
RFC 5795, [595](#)
RFID (*ver* identificação por radiofrequência)
Retroespalhamento RFID, [74](#)
Rede RFID, [73](#) -75
Rijmen, Vincent, [784](#)
Rijndael, [784](#) -787
Cifra Rijndael, [312](#)
Anel
pacote resiliente, [271](#)
token, [271](#)
Rivest, Ron, [773](#), 792, 795, 797, 804
Rivest Shamir Adleman algoritmo, [794](#) -796
RNC (*consulte* o controlador de rede de rádio)
Sinalização de bits roubados, [155](#)
Roberts, Larry, [56](#)
Compressão robusta de cabeçalho, [594](#)
ROHC (*veja* compressão de cabeçalho ROBust)
Servidor de nome raiz, [620](#)

AODV, [389](#)
Bellman-Ford, [370](#)
transmissão, [380](#) -382
baseado em classe, [421](#)
interdomínio sem classes, [447](#) - 449
protocolo multicast de vetor de distância, [383](#)
dinâmico, [364](#)
hierárquico, [378](#)-380
batata quente, [484](#)
interdomínio, [474](#)
internetwork, [431](#) -432
intradomínio, [474](#)
estado do link, [373](#) - 378
host móvel, [386](#) - 389
multicast, [382](#) - 385
multidestinação, [380](#)
camada de rede, [362](#) -392
cebola, [863](#)
OSPF, [464](#) -479
multicast de vetor de distância, [383](#)
qualidade de serviço, [415](#)
sessão, [362](#)
caminho mais curto, [366](#) -368
estático, [364](#)
ciente do tráfego, [395](#) - 396
triângulo, [388](#)
buraco de minhoca, [336](#)
Política de roteamento, [432](#)
RPC (*ver* chamada de procedimento remoto)
RPR (*ver* anel de pacote resiliente)
RRSet (*consulte* Conjunto de registros de recursos)
RSA (*consulte* o algoritmo Rivest Shamir Adleman)
RSVP (*consulte* Protocolo de reSerVação de recursos)
RTCP (*consulte* Protocolo de Controle de Transporte em Tempo Real)
RTO (*consulte* Retransmission TimeOut, TCP)
RTP (*consulte* Protocolo de Transporte em Tempo Real)
RTS (*ver* solicitação de envio)
RTSP (*consulte* Protocolo de transmissão em tempo real)
Codificação de comprimento de execução, [709](#)

S

S-box, criptográfico, [779](#)
S / MIME, [846](#)
SA (*consulte* Associação de Segurança)
SACK (*ver* reconhecimento seletivo)
Sandbox, [858](#)
Satélite
comunicação, [116](#) -125
geoestacionário, [117](#)
óbita terrestre baixa, [121](#) -123
óbita terrestre média, [121](#)
Pegada do satélite, [119](#)
Hub satélite, [119](#)
Sawtooth, TCP, [579](#)
Rede escalável, [34](#)
Scatternet, Bluetooth, [320](#)
Programação, pacote, [411](#) -414
Esquema, HTTP, [650](#)
SCO (*consulte* link orientado a conexão síncrona)
Scrambler, [128](#)
SCTP (*ver* protocolo de transmissão de controle de fluxo)
SDH (*consulte* Hierarquia digital síncrona)
Rede de telefonia móvel de segunda geração, [170](#) -174
Antena setorizada, [178](#)
DNS seguro, [850](#) -853
HTTP seguro, [853](#)
Nomenclatura segura, [848](#) -853
Bluetooth de emparelhamento seguro, [325](#)
Emparelhamento simples seguro, Bluetooth, [325](#)
Camada de soquetes seguros, [853](#) -857
Seguro / MIME, [846](#)
Segurança

Bluetooth, [826](#) -827
comunicação, [813](#) -827
enviar e-mail, [841](#) -846
IEEE 802.11, [823](#) -826
IP, [814](#) -818
Applet Java, [857](#) -858
código móvel, [857](#) -860
questões sociais, [860](#)-869
camada de transporte, [856](#)
Web, [856](#) -860
sem fio, [822](#) -827
Associação de segurança, [815](#)
Segurança por obscuridade, [768](#)
Principal de segurança, [773](#)
Ameaças de segurança, [847](#)-848
Seeder, BitTorrent, [751](#)
Segmento, [499](#), 542
Cabeçalho de segmento, TCP, [557](#) -560
Confirmação seletiva, TCP, [560](#), 580
Protocolo de repetição seletiva, [239](#) -244
Auto-similaridade, [737](#)

Página 952

928

ÍNDICE

Taxa de envio, regulamento, [535](#)-539
Janela de envio, [228](#)
Rede de sensores, [13](#) , 73-75
Protocolo de Internet de linha serial, [245](#)
Servidor, [4](#)
Farm de servidores, [64](#) , 738-741
Lado do servidor na Web, [655](#) -658
Geração do lado do servidor Web page, [673](#) -676
Stub do servidor, [544](#)
Serviço
orientado à conexão, [35](#) -38, 359-361
conexão, [35](#) -38, 358-359
Acordo de nível de serviço, [407](#)
Serviço primitivo, [38](#) - 40
Serviço
IEEE 802.11, [311](#) -312
integrado, [418](#) -421
fornecido pela camada de transporte, [495](#) -507
fornecido para a camada de transporte, [356](#) -357
em relação aos protocolos, [40](#)
Usuário de serviço, transporte, [497](#)
Servindo nó de suporte GPRS, [68](#)
Sessão, [44](#)
Protocolo de iniciação de sessão, [731](#) , 731-735
em comparação com o H.323, [733](#) -734
Chave de sessão, [828](#)
Camada de sessão, [44](#) -45
Roteamento de sessão, [362](#)
Decodificador, [4](#) , 723
SGSN (*consulte* Servindo Nó de Suporte GPRS)
SHA (*consulte* Secure Hash Algorithm)
Shannon, Claude, [94](#) -95
Limite de Shannon, 100, 106, [146](#)
Segredo compartilhado, autenticação usando, [828](#) -833
Espaçamento curto entre quadros, [308](#)
Serviço de mensagens curtas, [12](#)
Roteamento do caminho mais curto, [366](#) -368
SIFS (*ver* espaçoamento curto entre quadros)
Sinal, balanceado, [129](#)-130
Relação sinal-ruído, [94](#)
Sinalização
canal comum, [155](#)
in-band, [155](#)
bit roubado [155](#)

Bloco de assinatura, [629](#)
Assinaturas, digital, [797](#) - 806
Assinatura, código, [858](#)
Síndrome da janela boba, [567](#)
Cartão SIM, [69](#), 171
Protocolo de Internet simples, mais, [457](#)
Protocolo de transferência de correio simples, [625](#), 638-641
Protocolo de acesso a objetos simples, [682](#)
Link simplex, [97](#)
Fibra monomodo, [101](#)
Árvore afundar, [365](#)
SIP (ver Protocolo de Iniciação de Sessão)
SIPP (consulte Simple Internet protocol Plus)
Pele, jogador, [715](#)
SLA (ver Acordo de Nível de Serviço)
Janela deslizante, TCP, [565](#) –568
Protocolo de janela deslizante, 1 bit, [229](#) -232
[226-244](#), 229–232, 522
SLIP (ver protocolo de Internet de linha serial)
Slot, [264](#)
Fenda ALOHA, [264-266](#), 265
Início lento, TCP, [574](#)
limiar, [576](#)
Smartphone, [12](#)
Smiley, [623](#)
SMS (ver serviço de mensagens curtas)
SMTP (consulte Protocolo de transferência de correio simples)
Correio tradicional, [623](#)
SNR (ver relação sinal-ruído)
SOAP (consulte Simple Object Access Protocol)
Questões sociais, [14](#) -16
segurança, [860](#) -869
Rede social, [8](#)
Soquete, [59](#)
Berkeley, [500](#) -507
TCP, [553](#)
Programação de soquete, [503](#) –507
Transferência suave, [178](#)
Decodificação de decisão suave, [208](#)
Soliton, [103](#)
SONET (ver Synchronous Optical NETwork)
Porta de origem, [453](#)
Spam, [623](#)
Árvore de abrangência, [382](#)
Ponte de árvore de extensão, [337](#) - 340
Poema de Spanning Tree, [339](#)
SPE (consulte Synchronous Payload Envelope)
Alocação de espectro, [182](#) -183
Leilão de espectro, [112](#)
Velocidade da luz, [106](#)
Divisor, [149](#)
Spoofing, DNS, [848](#) -850
Feixe focal, [119](#)
Espalhamento de espectro, [135](#)
sequência direta, [108](#)
salto de frequência, [107](#)

Alocação de canal estático, [258](#) -261
Página estática, Web, [649](#)
Roteamento estático, [364](#)
Página da Web estática, [649](#), 662-663
Manutenção da estação, [118](#)
Multiplexação estatística, [34](#)
Multiplexação estatística por divisão de tempo, [135](#)
STDM (*consulte* Multiplexação por divisão de tempo estatística)
Esteganografia, [865](#) -867
Protocolo de parar e esperar, [221](#) -226, 522
Comutação de pacotes de armazenamento e encaminhamento, [36](#), 356
Modo de codificação de fluxo, [790](#) -791
Protocolo de transmissão de controle de fluxo, [503](#), 527
Streaming de áudio e vídeo, [697](#) -734
Streaming de mídia ao vivo, [721](#) -724
Mídia de streaming, [699](#)
Streaming de mídia armazenados, [713](#) -720
Engrenagem Strowger, [161](#)
Rede P2P estruturada, [754](#)
Transporte de fluxo estruturado, [503](#)
STS-1 (*consulte* Sinal de Transporte Síncrono-1)
Stub
cliente, [544](#)
servidor, [544](#)
Área de stub, [477](#)
Rede stub, [481](#)
Folha de estilo, [670](#)-671
Subcamada, controle de acesso médio, [257](#) -350
Sub-rede, [24](#), 444-446
Protocolo de sub-rede da Internet, [444](#) -446
Máscara de sub-rede, [443](#)
Sub-redes, [444](#)
Módulo de identidade de assinante, [69](#), 171
Cifra de substituição, [769](#)-770
Superframe, estendido, [154](#)
Supergrupo, [153](#)
Supernet, [447](#)
Swarm, BitTorrent, [751](#)
Switch, [24](#)
em comparação com ponte e hub, [340](#) -342
Ethernet, [20](#), 289
Ethernet comutada, [20](#), 280, 288-290
Troca, [161](#) -164
circuito, [161](#) -162
cut-through, [36](#), 336
camada de enlace de dados, [332](#) -349
etiqueta, [360](#), 470-474
mensagem, [600](#)
pacote, [162](#) -164
armazenar e encaminhar, [36](#)
Elemento de comutação, [24](#)
Símbolo, [126](#)
Taxa de símbolo, [127](#)
De chave simétrica criptografia, [778](#) -793
AES, [783](#) -787
modo de feedback de cifra, [789](#) - 790
modo de contador, [791](#) -792
DES, [780](#) -782
modo de livro de código eletrônico, [787](#) -788
Rijndael, [784](#) -787
modo de cifra de fluxo, [790](#) -791
DES triplo, [782](#) -783
Assinatura de chave simétrica, [798](#) - 799
Cookie SYN, TCP, [561](#)
Ataque de inundação SYN, [561](#)
Sincronização, [44](#)
Link orientado para conexão síncrona, [325](#)
Hierarquia digital síncrona, [156](#) –159
Rede ótica síncrona, [156](#) –159
Envelope de carga útil síncrona, [157](#)
Sinal de transporte síncrono-1, [157](#)

Sistema, distribuído, [2](#)
Código sistemático, [204](#)

T

Portadora T1, [154](#)–155
Linha T1, [128](#), 154
Tag, HTML, [663](#) –666
Troca de etiqueta, [471](#)
Queda de cauda, [412](#)
Talkspurt, [551](#)
Escritório tandem, [141](#)
TCG (*consulte* Trusted Computing Group)
TCM (*consulte* Modulação Codificada em Trellis)
TCP (*consulte* Protocolo de Controle de Transmissão)
TDD (*veja* Duplex de Divisão de Tempo)
TDM (*consulte* Multiplexação por Divisão de Tempo)
Telco, [61](#)

Página 954

930

ÍNDICE
Telefone
sem fio, [165](#)
móvel, [164](#) -179
inteligente, [12](#)
Sistema telefônico, [138](#) -164
estação final, [140](#)
banda de guarda, [133](#)
tempo de guarda, [135](#)
loop local, [144](#)-152
móvel, [164](#) -179
modem, [145](#)
modulação, [130](#) -132
ponto de presença, [143](#)
política, [142](#) -144
estrutura, [139](#) -142
comutação, [161](#) -164
escritório tandem, [141](#)
pedágio, [141](#)
tronco, [152](#) -160
Tronco de telefone, [152](#) -160
Televisão
cabô, [179](#) -186
antena comunidade, [179](#) -180
Mascaramento temporal, [703](#)
Protocolo de integridade de chave temporária, [825](#)
Terminal, VoIP, [728](#)
Mensagens de texto, [12](#)
Mensagens de texto, [12](#)
Projeto de Parceria de Terceira Geração, [76](#)
Terceira geração de rede de telefonia móvel, [65](#) -69,
[174](#) –179
Cookie da Web de terceiros, [662](#)
Ameaças, segurança, [847](#)-848
Problema de três ursos, [450](#)
Aperto de mão de três vias, [516](#)
ISP Tier 1, [64](#)
Rede Tier 1, [437](#)
Duplex por divisão de tempo, [316](#) -317
Multiplexação por divisão de tempo, [135](#) , 154-156
Horário, [135](#)
Gestão Timer, TCP, [568](#) -571
Timestamp, TCP, [560](#)
Roda dentada, [593](#)
Estratégia tit-for-tat, BitTorrent, [752](#)
TKIP (*consulte* Protocolo de Integridade de Chave Temporária)
TLS (*consulte* Segurança da Camada de Transporte)
Token, [271](#)
Algoritmo de token bucket, [397](#) , 408-411
Barramento de token, [272](#)

Gerenciamento de token, [44](#)
Protocolo de passagem de fichas, [271](#) -272
Token ring, [271](#)
Tronco de conexão com pedágio, [141](#)
Pedágio, [141](#)
Domínio de nível superior, [613](#)
Torrent, BitTorrent, [750](#)
TPDU (*consulte Unidade de dados do protocolo de transporte*)
TPM (*consulte Módulo de plataforma confiável*)
Traceroute, [466](#)
Tracker, BitTorrent, [751](#)
Análise de tráfego, [815](#)
Engenharia de tráfego, [396](#)
Polícia de trânsito, [407](#)
Traffic shaping, [407](#) , 407-411
Estrangulamento do tráfego, [398](#) -401
Tráfego-aware roteamento, [395](#) -396
Trailer, [32](#) , 194, 216, 250, 326
Transcodificação, [694](#)
Agente de transferência, [624](#)-625, 630-631
Serviço de trânsito, [480](#)
Transmissão
banda base, [125](#)
luz, [14](#) -116
banda passante, [125](#)
sem fios, [105](#) -116
Protocolo de controle de transmissão (TCP), [47](#) , 552-582
relógio de reconhecimento, [574](#)
camada de aplicação, [47](#) -48
comparação com OSI, [49](#) -51
colapso do congestionamento, [572](#)
controle de congestionamento, [571](#) -581
janela de congestionamento, [571](#)
estabelecimento de conexão, [560](#) -562
gerenciamento de conexão, [562](#) -565
liberação de conexão, [562](#)
crítica, [53](#) -54
reconhecimento cumulativo, [558](#), 568
reconhecimento atrasado, [566](#)
confirmação duplicada, [577](#)
recuperação rápida, [578](#)
retransmissão rápida, [577](#)
futuro, [581](#) -582
introdução, [552](#) -553
Algoritmo de Karn, [571](#)
cronômetro de manutenção de atividade, [571](#)
camada de link, [46](#)
tamanho máximo do segmento, [559](#)
unidade máxima de transferência, [556](#)
Algoritmo de Nagle, [566](#)
desempenho, [582](#) -599

ÍNDICE

931

Protocolo de controle de transmissão (*continuação*)
temporizador de persistência, [571](#)
porta, [553](#)
modelo de referência, [45](#) -51
tempo limite de retransmissão, [568](#)
dente de serra, [579](#)
cabeçalho do segmento, [557](#) -560
reconhecimento seletivo, [580](#)
síndrome da janela boba, [567](#)
janela deslizante, [565](#) -568
início lento, [574](#)
limite de início lento, [576](#)
soquete, [553](#)
acelerando, [582](#) -599

Cookie SYN, [561](#)
Ataque de inundação SYN, [561](#)
gestão timer, [568](#) -571
opção de carimbo de data / hora, [560](#)
camada de transporte, [47](#)
dados urgentes, [555](#)
porto conhecido, [553](#)
sonda de janela, [566](#)
escala da janela, [560](#)
Linha de transmissão, [24](#)
Meios de transmissão, guiado, [85](#) -105
Oportunidade de transmissão, [309](#)
Controle de transmissão de energia, IEEE 802.11, [312](#)
Transponder, [116](#)
Transporte, fluxo estruturado, [503](#)
Entidade de transporte, [496](#)
Camada de transporte, [44](#)
endereçamento, [509](#) -512
controle de congestionamento [530](#)-541
rede tolerante a atrasos, [599](#) -605
controle de erro [522](#)-527
controle de fluxo, [522](#) -527
desempenho, [582](#) -599
porto, [509](#)
segurança, [856](#)
TCP, [552](#) -582
TCP / IP, [47](#)
Protocolos de transporte, [507](#) -530
serviço de transporte [495](#) -507
UDP, [541](#) -552
Modo de transporte, segurança IP, [815](#)
Protocolo de transporte [507](#) - 530, [541](#) - 582
questões de design, [507](#) -530
Unidade de dados de protocolo de transporte, [499](#)
Ponto de acesso do serviço de transporte, [509](#)
Primitivo de serviço de transporte [498](#) - 500
Provedor de serviço de transporte, [497](#)
Usuário do serviço de transporte, [497](#)
Cifra de transposição [771](#)-772
Protocolo de caminhada na árvore, adaptativo [275](#)-277
Modulação codificada em treliça, [146](#)
Roteamento de triângulo, [388](#)
Trigrama, [70](#)
DES triplo, [782](#) -783
Tronco, telefone, [152](#) -160
Âncora de confiança, [812](#)
Computação confiável, [869](#)
Módulo de plataforma confiável, [869](#)
TSAP (*consulte* Ponto de Acesso do Serviço de Transporte)
Modo túnel, IPSec, [815](#)
Tunelamento, [387](#) , 429-431
Par trançado, [96](#) -97
sem blindagem, [97](#)
Twitter, [8](#)
Problema Two-exército, [518](#) -519
TXOP (*ver* oportunidade de transmissão)

você

U-NII (*ver* Informação Nacional Não Licenciada
A infraestrutura)
Computação ubíqua, [10](#)
UDP (*ver* protocolo de datagrama do usuário)
UHF RFID, [73](#) -74
Banda ultralarga, [108](#)
UMTS (*ver* Telecomunicações Móveis Universais
Sistema)
Ponto não chocado, BitTorrent, [752](#)
Unicast, [385](#)
Unicast, [17](#) , 385
Identificador de recurso uniforme, [652](#)
Localizador uniforme de recursos, [650](#)
Nome de recurso uniforme, [652](#).

Sistema universal de telecomunicações móveis, [65](#), 175
Barramento serial universal, [128](#)
Infraestrutura de informação nacional não licenciada, [113](#)
Par trançado sem blindagem, [97](#)
Rede P2P não estruturada, [754](#)
Proxy upstream, [742](#)
Dados urgentes, [555](#)
URI (*consulte* Identificador Uniforme de Recursos)
URL (*ver* esquema)
URN (*ver* nome de recurso uniforme)
USB (*consulte* Universal Serial Bus)

Página 956

932

ÍNDICE

Usuário, celular, [10 a](#) 13
Agente do usuário, [624](#), 626
Protocolo de datagrama do usuário, [47](#), 541-552, 542, [549](#) -550
porta, [542](#)
transmissão em tempo real [546](#) -552
chamada de procedimento remoto, [541](#) -543
RTP, [547](#) -549
Protocolo Utopia, [220](#) –222
UTP (*ver* par trançado não blindado)
UWB (*consulte* Ultra-WideBand)

V

Modem V.32, [146](#)
Modem V.34, [146](#)
Modem V.90, [147](#)
Modem V.92, [147](#)
Agente de férias, [629](#)
Torneira de vampiro, [291](#)
Cinto Van Allen, [117](#)
VC (*ver* Circuito Virtual)
Terminal de abertura muito pequena, [119](#)
Vídeo
entrelaçado, [705](#)
progressivo, [705](#)
streaming, [704](#) -712
Compressão de vídeo, [706](#)
Campo de vídeo, [705](#)
Vídeo sob demanda, [713](#)
Servidor de vídeo, [414](#), 416
Circuito virtual, [249](#), 358-361
Rede de circuito virtual, [358](#)
comparação com rede de datagramas, [361](#) -362
Virtual LAN, [21](#), 332, 342-349
Rede privada virtual, [4](#), 26, 431, 821-822
Rede de circuito virtual, [358](#)
Vírus, [860](#)
Registro de localização de visitante, [171](#)
VLAN (*consulte* LAN Virtual)
VLR (*ver* Registro de localização do visitante)
Trato vocal, [702](#)
Vocoder, [701](#)
VOD (*veja o* vídeo sob demanda)
Voz sobre IP, [5](#), 36, 698, 725, 728-734
Sinais de voz, digitalização, [153](#)-154
Linha de grau de voz, [93](#)
VoIP (*consulte* Voz sobre IP)
VPN (*consulte* Rede privada virtual)
VSAT (*consulte* Terminal de Abertura Muito Pequena)

W

W3C (*consulte* World Wide Web Consortium)
Jardim murado, [723](#)
Código Walsh, [136](#)
WAN (*consulte* Wide Area Network)

WAP (*consulte* Protocolo de aplicativo sem fio)
Marca d'água, [867](#)
Codificação de forma de onda, [702](#)
Comprimento de onda, [106](#)
Multiplexação por divisão de comprimento de onda, [159](#) -160
WCDMA (*ver* Wideband Code Division
Acesso múltiplo)
WDM (*ver* Wavelength Division Multiplexing)
Computador vestível, [13](#)
Web (*ver* World Wide Web)
Aplicativo da web, [4](#)
Navegador da web, [648](#)
extensão, [859](#) -860
aplicativo auxiliar, [654](#)
plug-in, [653](#) -654, 859
proxy, [741](#) -742
Webmail, [645](#), 645-646
Fila justa ponderada, [414](#)
Porta conhecida, TCP, [553](#)
WEP (*consulte* Privacidade equivalente com fio)
WFQ (*consulte* Weighted Fair Queuing)
Espaço em branco, [113](#)
Clareamento, [781](#)
Rede [de](#) longa distância, [23](#) , 23-27
Acesso múltiplo por divisão de código de banda larga, [65](#) , 175
WiFi (*consulte* IEEE 802.11)
WiFi Alliance, [76](#)
Acesso protegido por WiFi, [73](#) , 311
Acesso protegido por WiFi [2](#)
[312](#) , 823
Wiki, [8](#)
Wikipedia, [8](#)
WiMAX (*consulte* IEEE 802.16)
WiMAX Forum, [313](#)
Sonda de janela, TCP, [566](#)
Escala da janela, TCP, [560](#)
Vinho, política de redução de carga, [401](#)
Privacidade equivalente com fio, [73](#) , 311, 823

Página 957

ÍNDICE
933
Sem fio
banda larga, [312](#) -320
fixo, [11](#)
Protocolo de aplicativo sem fio, [693](#)
Problemas sem fio, [539](#)-541
LAN sem fio, [39](#) , 277-280, 299-312
LAN sem fio (*consulte* IEEE 802.11)
Protocolo de LAN sem fio, [277](#) -280
Protocolos LAN sem fios, [277](#) -280
Roteador sem fio, [19](#)
Segurança sem fio, [822](#) -827
A transmissão sem fio, [105](#) -116
Fator de trabalho, criptográfico, [768](#)
World Wide Web, [2](#) , 646-697
AJAX, [679](#) -683
visão geral da arquitetura, [647](#) -649
cache, [690](#) -692
folhas de estilo em cascata, [670](#)-672
lado do cliente, [649](#) -652
geração de página do lado do cliente, [676](#)-678
conexões, [684](#) -686
biscoitos, [658](#) -662
rastrejando, [696](#)
páginas dinâmicas, [672](#)
formulários, [667](#) -680
HTML, [663](#) - 667
HTTP, [683](#) -684

cabeçalhos de mensagens, [688](#) - 690
métodos, [686](#) -688
Tipos MIME, [652](#) -655
web móvel, [693](#) -695
página, [647](#)
proxy, [692](#), 741-743
pesquisa, [695](#) -697
segurança [856](#) -860
rastreamento, [661](#)
lado do servidor, [655](#) -658
geração da página do lado do servidor, [673](#) -676
páginas estáticas, [662](#) -672
World Wide Web Consortium, [82](#), 647
Roteamento de buraco de minhoca, [336](#)
WPA (*consulte* Acesso protegido por WiFi)
WPA2 (*consulte* Acesso protegido por WiFi 2)

X

Padrão X.400, [629](#)
X.509, [809](#) –810
XHTML (veja eXtended HyperText Markup Língua)
XML (consulte eXtensible Markup Language)
XSLT (consulte eXtensible Stylesheet Language Transformação)

Z

Lei de Zipf, [737](#)
Zona
DNS, [619](#) -620
multimídia, [728](#)

Página 958

Também por Andrew S. Tanenbaum

Modern Operating Systems, 3^a ed.

Este best-seller mundial incorpora os mais recentes desenvolvimentos em sistemas operacionais. o livro começa com capítulos sobre os princípios, incluindo processos, gerenciamento de memória, sistemas de arquivos, E / S e assim por diante. Em seguida, vai para estudos de caso de três capítulos: Linux, Windows e Symbian. A experiência de Tanenbaum como designer de três sistemas operacionais (Amoeba, Globe e MINIX) dá a ele um histórico que poucos outros autores podem igualar, então o capítulo final destila sua longa experiência em conselhos para designers de sistemas operacionais.

Página 959

Também por Andrew S. Tanenbaum e Albert S. Woodhull

Sistemas Operacionais: Design e Implementação, 3^a ed.

Todos os outros livros sobre sistemas operacionais são longos em teoria e curtos em prática. Esta está diferente. Além do material usual sobre processos, gerenciamento de memória, sistemas de arquivos, E / S, e assim por diante, contém um CD-ROM com o código-fonte (em C) de um pequeno, mas completo, POSIX-sistema operacional compatível chamado MINIX 3 (consulte www.minix3.org) Todos os princípios são ilustrados mostrando como eles se aplicam ao MINIX 3. O leitor também pode compilar, testar e experimentar com o MINIX 3, levando a um conhecimento profundo de como um sistema operacional realmente funciona.

Página 960

Também por Andrew S. Tanenbaum

Structured Computer Organization, 5^a ed.

Um computador pode ser estruturado como uma hierarquia de níveis, desde o hardware até o sistema operacional. Este livro trata de todos eles, começando com como um transistor funciona e terminando com design do sistema operacional. Nenhuma experiência anterior com hardware ou software é necessária para seguir neste livro, no entanto, como todos os tópicos são independentes e explicados em termos simples, começando logo no início. Os exemplos de corrida usados em todo o livro vão desde os mais sofisticados UltraSPARC III, do sempre popular x86 (Pentium) ao minúsculo Intel 8051 usado em pequenas embarcações sistemas dedicados.

Página 961

Também por Andrew S. Tanenbaum e Maarten van Steen

Sistemas Distribuídos: Princípios e Paradigmas, 2^a ed.

Os sistemas distribuídos estão se tornando cada vez mais importantes no mundo e este livro explica seus princípios e os ilustra com numerosos exemplos. Entre os tópicos abordados estão os arquivos tecturas, processos, comunicação, nomenclatura, sincronização, consistência, tolerância a falhas e segurança ridade. Os exemplos são retirados de sistemas baseados em objetos distribuídos, arquivos, baseados na Web e baseados em coordenação tems.

Página 962

SOBRE OS AUTORES

Andrew S. Tanenbaum é graduado em SB pelo MIT e Ph.D. da Universidade de Califórnia em Berkeley. Atualmente é Professor de Ciéncia da Computação na Vrije Universiteit onde ensinou sistemas operacionais, redes e tópicos relacionados por mais de 30 anos. Sua corrente pesquisa é em sistemas operacionais altamente confiáveis, embora ele tenha trabalhado em compiladores, distribuídos sistemas, segurança e outros tópicos ao longo dos anos. Esses projetos de pesquisa levaram a mais de 150 referências artigos publicados em periódicos e conferências.

O Prof. Tanenbaum também (co) escreveu cinco livros que já apareceram em 19 edições.

Os livros foram traduzidos para 21 idiomas, que vão do basco ao tailandês, e são usados em universidades versidades em todo o mundo. Ao todo, são 159 versões (combinações de idioma / edição), que estão listados em www.cs.vu.nl/~ast/publications.

O Prof. Tanenbaum também produziu um volume considerável de software, incluindo Amsterdã dam Compiler Kit (um compilador portátil retargetable), Amoeba (um sistema distribuído antigo usado em LANs) e Globe (um sistema distribuído de área ampla).

Ele também é o autor do MINIX , um pequeno clone UNIX inicialmente planejado para uso em programas de estudantes laboratórios de gramática. Foi a inspiração direta para o Linux e a plataforma na qual o Linux foi inicialmente desenvolvido. A versão atual do MINIX , chamada MINIX 3 , agora está focada em ser extremamente re-sistema operacional responsável e seguro. O Prof. Tanenbaum irá considerar seu trabalho feito quando não houver O computador está equipado com um botão de reinicialização e nenhuma pessoa viva experimentou uma pane no sistema. MINIX 3 é um projeto de código aberto em andamento para o qual você está convidado a contribuir. Vamos para www.minix3.org para baixar uma cópia gratuita e descobrir o que está acontecendo.

Tanenbaum é Fellow da ACM, Fellow do IEEE e membro do Royal

Academia Holandesa de Artes e Ciéncias. Ele também ganhou vários prêmios científicos, incluindo:

Prêmio TAA McGuffey de 2010 para livros de Ciéncia da Computação e Engenharia

2007 IEEE James H. Mulligan, Jr. Medalha de Educação

Prêmio TAA Texty 2002 para livros de Ciéncia da Computação e Engenharia

Prêmio ACM / SIGCSE de 1997 por Contribuições Destacadas para a Educação em Ciéncia da Computação

Prêmio ACM Karl V. Karlstrom de Educador de Destaque de 1994

Sua home page na World Wide Web pode ser encontrada em <http://www.cs.vu.nl/~ast/>.

David J. Wetherall é Professor Associado de Ciéncia da Computação e Engenharia da Universidade cidade de Washington em Seattle e consultor do Intel Labs em Seattle. Ele vem da Austrália, onde ele recebeu seu BE em engenharia elétrica pela University of Western Australia e seu Ph.D. em ciéncia da computação do MIT

O Prof. Wetherall trabalhou na área de redes nas últimas duas décadas. Sua pesquisa é focado em sistemas de rede, especialmente redes sem fio e computação móvel, o design do In-protocolos ternet e medição de rede.

Ele recebeu o prêmio ACM SIGCOMM Test-of-Time pela pesquisa que foi pioneira na rede ativa funciona, uma arquitetura para a introdução rápida de novos serviços de rede. Ele recebeu o IEEE William Prêmio Bennett por avanços em mapeamento da Internet. Sua pesquisa foi reconhecida com um NSF Prêmio CAREER em 2002, e ele se tornou um Sloan Fellow em 2004.

Além de ensinar redes, o Prof. Wetherall participa da pesquisa de redes comunidade. Ele co-presidiu os comitês de programa da SIGCOMM, NSDI e MobiSys, e co-fundou as oficinas ACM HotNets. Ele atuou em vários comitês de programa para conferências de trabalho e é editor da ACM Computer Communication Review.

Sua home page na World Wide Web pode ser encontrada em <http://djw.cs.washington.edu> .