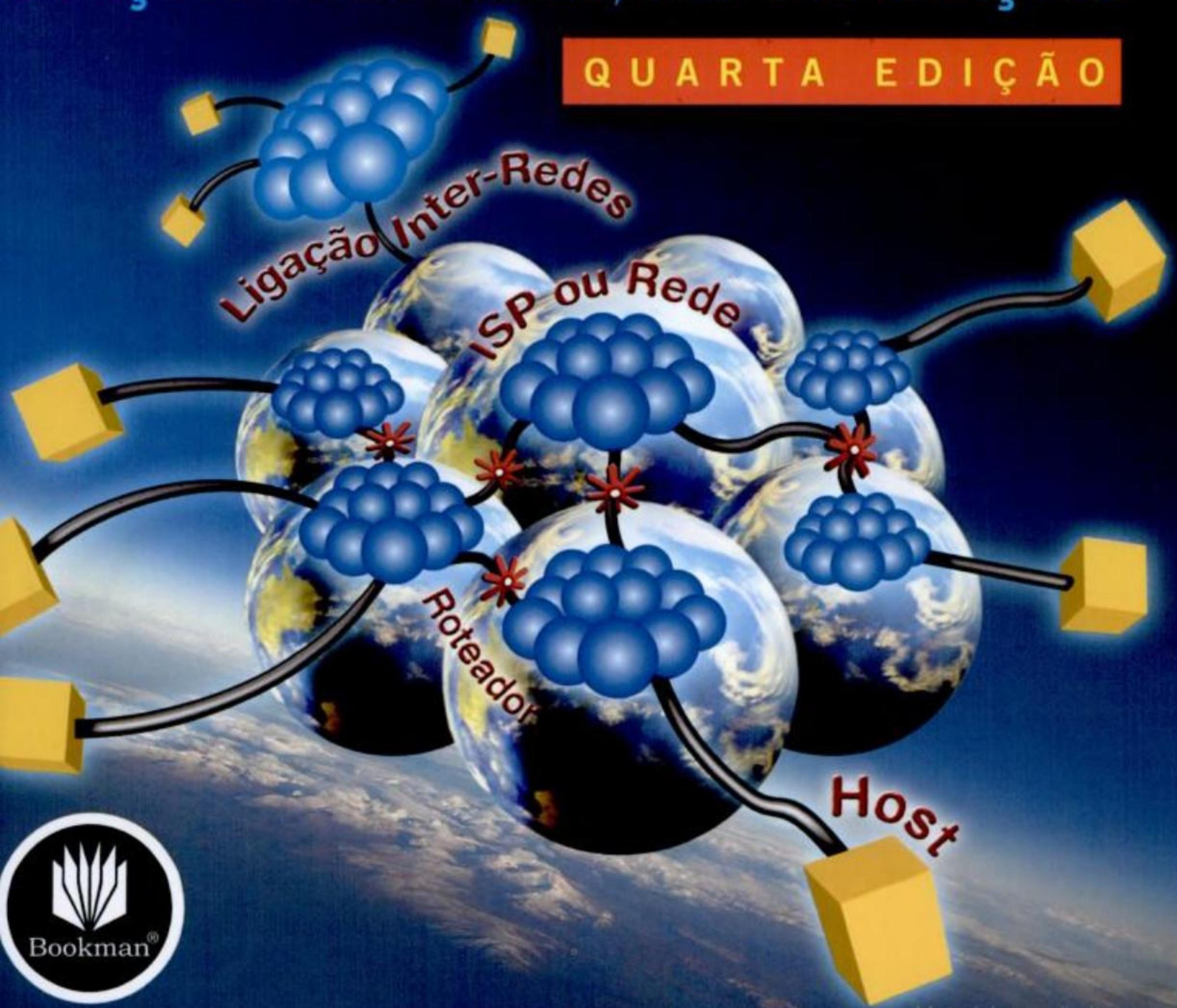


Douglas E. Comer

REDES DE COMPUTADORES E INTERNET

Abrange TRANSMISSÃO de DADOS,
LIGAÇÕES INTER-REDES, WEB e APLICAÇÕES

QUARTA EDIÇÃO



Authorized translation from the English language edition, entitled COMPUTER NETWORKS AND INTERNETS WITH INTERNET APPLICATIONS, 4th Edition by COMER, DOUGLAS E.; DROMS, RALPH, E., published Pearson Education, Inc., publishing as Prentice Hall, Copyright © 2004. All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

Portuguese language edition published by Bookman Companhia Editora Ltda, a Division of Artmed Editora SA, Copyright © 2007.

Tradução autorizada a partir do original em língua inglesa da obra intitulada COMPUTER NETWORKS AND INTERNETS WITH INTERNET APPLICATIONS, 4^a Edição de autoria de COMER, DOUGLAS E.; DROMS, RALPH, E., publicado por Pearson Education, Inc., sob o selo de Prentice Hall, Copyright © 2004. Todos os direitos reservados. Este livro não poderá ser reproduzido nem em parte nem na íntegra, nem ter partes ou sua íntegra armazenado em qualquer meio, seja mecânico ou eletrônico, inclusive fotoreprografia, sem permissão da Pearson Education, Inc.

A edição em língua portuguesa desta obra é publicada por Bookman Companhia Editora Ltda, uma divisão da Artmed Editora SA, Copyright © 2007.

ISBN 0-13-143351-2

Capa: *Gustavo Macrì*, arte sobre capa original

Leitura final: *Cristina Foresti Piccoli*

Supervisão editorial: *Denise Weber Nowaczyk*

Editoração eletrônica: *Laser House*

O CD-ROM é fornecido como está, no idioma original em inglês e sem custo adicional. As editoras Pearson Education Inc. e Bookman Companhia Editora não se responsabilizam por perdas e danos causados pelo mau uso do conteúdo.

UNIX é marca registrada de The Open Group nos Estados Unidos e em outros países. Microsoft Windows, Windows 95, Windows 98 e Windows NT são marcas registradas de Microsoft Corporation. Microsoft é marca registrada de Microsoft Corporation. Solaris, Sparc, Java e JavaScript são marcas registradas de Sun Microsystems, Incorporated. Sniffer é marca registrada de Network General Corporation. Ada Magic é marca registrada de Intermetrics, Incorporated. Alpha é marca registrada de Digital Equipment Corporation. Pentium é marca registrada de Intel Corporation. X Window System é marca registrada de X Consortium, Incorporated. Smartjack é marca registrada de Westell, Incorporated.

Reservados todos os direitos de publicação, em língua portuguesa, à
ARTMED® EDITORA S. A.

(BOOKMAN® COMPANHIA EDITORA é uma divisão da ARTMED® EDITORA S.A.)

Av. Jerônimo de Ornelas, 670 - Santana
90040-340 Porto Alegre RS
Fone (51) 3027-7000 Fax (51) 3027-7070

É proibida a duplicação ou reprodução deste volume, no todo ou em parte,
sob quaisquer formas ou por quaisquer meios (eletrônico, mecânico, gravação,
fotocópia, distribuição na Web e outros), sem permissão expressa da Editora.

SÃO PAULO
Av. Angélica, 1091 - Higienópolis
01227-100 São Paulo SP
Fone (11) 3665-1100 Fax (11) 3667-1333

SAC 0800 703-3444

IMPRESSO NO BRASIL
PRINTED IN BRAZIL

Sumário

Usando e Construindo Aplicações Internet

Capítulo 1 Introdução 31

- 1.1 Crescimento das Redes de Computadores 33**
- 1.2 Complexidade em Sistemas de Rede 34**
- 1.3 Dominando a Complexidade 34**
- 1.4 Conceitos e Terminologia 34**
- 1.5 O Valor da Experiência Prática 34**
- 1.6 Organização do Livro 35**
- 1.7 Resumo 35**

Capítulo 2 Motivação e Ferramentas 36

- 2.1 Introdução 37**
- 2.2 Compartilhamento de Recursos 37**
- 2.3 Crescimento da Internet 38**
- 2.4 Testando a Internet 39**
- 2.5 Interpretando uma Resposta do Ping 41**
- 2.6 Traçando uma Rota 43**
- 2.7 Resumo 44**

Capítulo 3 Aplicativos e Programação em Rede 45

- [3.1 Introdução 47](#)
- [3.2 Comunicação em Rede 47](#)
- [3.3 Arquitetura Cliente-Servidor 48](#)
- [3.4 Paradigma da Comunicação 48](#)
- [3.5 Um Exemplo de API \(Application Program Interface\) 48](#)
- [3.6 Uma Olhada Intuitiva na API 49](#)
- [3.7 Definição de API 49](#)
- [3.8 Código para um Aplicativo de Eco 52](#)
- [3.9 Código Para Um Aplicativo de Chat 56](#)
- [3.10 Código para um Aplicativo de Web 60](#)
- [3.11 Gerenciando Múltiplas Conexões com a Função Select 66](#)
- [3.12 Resumo 66](#)

Meios de Transmissão

Capítulo 4 Meios de Transmissão 69

- [4.1 Introdução 71](#)
- [4.2 Fios de Cobre 71](#)
- [4.3 Fibras de Vidro 73](#)
- [4.4 Rádio 73](#)
- [4.5 Satélites 74](#)
- [4.6 Satélites Geossíncronos 75](#)
- [4.7 Satélites de Órbita Baixa da Terra 75](#)
- [4.8 Arrays de Satélites de Órbita Baixa da Terra 75](#)
- [4.9 Microonda 76](#)
- [4.10 Infravermelho 76](#)
- [4.11 Luz de Laser 76](#)
- [4.12 Resumo 77](#)

Capítulo 5 Comunicação Local Assíncrona (RS-232) 78

- [5.1 Introdução 79](#)
- [5.2 A Necessidade de Comunicação Assíncrona 79](#)
- [5.3 Usando Corrente Elétrica para Enviar Bits 80](#)
- [5.4 Padrões de Comunicação 80](#)
- [5.5 Taxa de Baud, Enquadramento e Erros 82](#)

5.6 Comunicação Assíncrona Full Duplex	83
5.7 Limitações do Hardware Real	84
5.8 Largura de Banda do Hardware e a Transmissão de Bits	85
5.9 O Efeito do Ruído na Comunicação	85
5.10 Importância para a Comunicação de Dados	86
5.11 Resumo	86

Capítulo 6 Comunicação de Longa Distância (Portadoras, Modulação e Modems) 88

6.1 Introdução	89
6.2 Enviando Sinais Através de Longas Distâncias	89
6.3 Hardware do Modem Usado para a Modulação e a Demodulação	92
6.4 Circuitos Alugados de Dados Analógicos (Leased Analog Data Circuits)	92
6.5 Freqüência Óptica, de Rádio e Modems Dial-up	93
6.6 Freqüências e Multiplexação da Portadora	95
6.7 Banda Base e Tecnologias de Banda Larga	96
6.8 Multiplexação por Divisão de Onda	96
6.9 Espectro Espalhado (Spread Spectrum)	96
6.10 Multiplexação por Divisão de Tempo	97
6.11 Resumo	97

Transmissão de Pacotes

Capítulo 7 Pacotes, Quadros e Detecção de Erro 99

7.1 Introdução	101
7.2 O Conceito de Pacotes	101
7.3 Pacotes e Multiplexação por Divisão de Tempo	103
7.4 Pacotes e Quadros de Hardware	103
7.5 Byte Stuffing	104
7.6 Erros de Transmissão	105
7.7 Bits de Paridade e Verificação de Paridade	106
7.8 Probabilidade, Matemática e Detecção de Erros	107
7.9 Detectando Erros com Checksums	108
7.10 Detectando Erros com Verificação de Redundância Cíclica	109
7.11 Combinando Blocos Básicos	110
7.12 Erros de Rajadas	111

[**7.13 Formato de Quadro e Mecanismos de Detecção de Erro 111**](#)

[**7.14 Resumo 112**](#)

Capítulo 8 *Tecnologias de LAN e Topologias de Rede 114*

[**8.1 Introdução 115**](#)

[**8.2 Comunicação Direta Ponto a Ponto 115**](#)

[**8.3 Canais de Comunicação Compartilhados 117**](#)

[**8.4 Importância das LANs e Localidade de Referência 118**](#)

[**8.5 Topologias de LAN 118**](#)

[**8.6 Exemplo de Rede de Barramento: Ethernet 120**](#)

[**8.7 Detecção de Portadora em Redes de Acesso Múltiplo
\(Carrier Sense on Multi-Access Networks, CSMA\) 122**](#)

[**8.8 Detecção de Colisões e Backoff com CSMA/CD 123**](#)

[**8.9 LANs Sem Fio 802.11b e CSMA/CA 124**](#)

[**8.10 Outro Exemplo de Rede de Barramento 125**](#)

[**8.11 Topologia em Anel e Passagem de Token 125**](#)

[**8.12 Redes de Passagem de Token de Autocura 127**](#)

[**8.13 Exemplo de Rede Estrela: ATM 128**](#)

[**8.14 Resumo 129**](#)

Capítulo 9 *Endereçamento de Hardware e Identificação de Tipo de Quadro 132*

[**9.1 Introdução 133**](#)

[**9.2 Especificando um Receptor 133**](#)

[**9.3 Como o Hardware de LAN Usa Endereços para Filtrar Pacotes 134**](#)

[**9.4 Formato de um Endereço Físico 135**](#)

[**9.5 Broadcasting 137**](#)

[**9.6 Multicasting 137**](#)

[**9.7 Endereçamento Multicast 138**](#)

[**9.8 Identificando o Conteúdo de Pacotes 139**](#)

[**9.9 Cabeçalhos e Formatos de Quadro 139**](#)

[**9.10 Um Exemplo de Formato de Quadro 140**](#)

[**9.11 Usando Redes que Não Têm Quadros Auto-Identificados 140**](#)

[**9.12 Analisadores de Rede, Endereços Físicos, Tipos de Quadro 143**](#)

[**9.13 Resumo 144**](#)

Capítulo 10 Cabeamento de LAN, Hardware de Topologia e Interface Físicos 146

- 10.1 Introdução 147**
- 10.2 Velocidades de LANs e Computadores 147**
- 10.3 Hardware de Interface de Rede 148**
- 10.4 A Conexão entre uma NIC e uma Rede 150**
- 10.5 Cabeamento Original Espesso de Ethernet 150**
- 10.6 Multiplexação de Conexão 151**
- 10.7 Cabeamento Fino de Ethernet 152**
- 10.8 Ethernet de Par Trançado 153**
- 10.9 Vantagens e Desvantagens de Esquemas de Cabeamento 154**
- 10.10 O Paradoxo da Topologia 155**
- 10.11 Placas de Interface de Rede e Esquemas de Cabeamento 155**
- 10.12 Interfaces de Rede 10/100 e Auto-Negociação 157**
- 10.13 Categorias de Cabos 158**
- 10.14 Esquemas de Cabeamento e Outras Tecnologias de Rede 158**
- 10.15 Resumo 159**

Capítulo 11 Estendendo LANs: Modems de Fibra, Repetidores, Bridges e Switches 161

- 11.1 Introdução 163**
- 11.2 Limitação de Distância e Projeto de LANs 163**
- 11.3 Extensões de Fibra Óptica 164**
- 11.4 Repetidores 165**
- 11.5 Bridges 167**
- 11.6 Filtragem de Quadros 168**
- 11.7 Comportamento Inicial e Estado Estável de Redes com Bridges 168**
- 11.8 Planejamento de uma Rede com Bridges 169**
- 11.9 Ligação com Bridge entre Edifícios 170**
- 11.10 Uso de Bridges Através de Distâncias Mais Longas 170**
- 11.11 Um Ciclo de Bridges 172**
- 11.12 Distributed Spanning Tree (Árvore de Extensão Distribuída) 173**
- 11.13 Comutação (Switching) 173**
- 11.14 Combinando Switches e Hubs 174**
- 11.15 Ligação por Bridges e Comutação com Outras Tecnologias 175**
- 11.16 Resumo 175**

Capítulo 12 *Tecnologias para Conexões Digitais de Longa Distância* 177

- 12.1 Introdução 179**
- 12.2 Telefonia Digital 179**
- 12.3 Comunicação Síncrona 181**
- 12.4 Circuitos Digitais, NIUs e DSU/CSUs 181**
- 12.5 Padrões Telefônicos 182**
- 12.6 Terminologia DS e Taxas de Dados 183**
- 12.7 Circuitos de Capacidade Mais Baixa 184**
- 12.8 Circuitos Digitais de Capacidade Intermediária 184**
- 12.9 Circuitos de Capacidade Mais Alta 185**
- 12.10 Padrões para Portadoras Ópticas 185**
- 12.11 O Sufixo C 185**
- 12.12 Synchronous Optical NETwork (SONET) 186**
- 12.13 O Loop de Assinante Local 187**
- 12.14 ISDN 187**
- 12.15 Tecnologia da Linha Assimétrica Digital de Assinante 188**
- 12.16 Outras Tecnologias de DSL 190**
- 12.17 Tecnologia de Modem de Cabo 191**
- 12.18 Comunicação Upstream 192**
- 12.19 Coaxial de Fibra Híbrido 193**
- 12.20 Fibra para o Meio-Fio (Curb) 194**
- 12.21 Modems de Head-End e Tail-End 194**
- 12.22 Alternativas Sem Fio para Casos Especiais 194**
- 12.23 Sistemas de Transmissão por Satélite 195**
- 12.24 Resumo 196**

Capítulo 13 *Tecnologias de WAN e Roteamento* 198

- 13.1 Introdução 199**
- 13.2 Redes Grandes e Áreas Extensas 199**
- 13.3 Switches de Pacotes 200**
- 13.4 Formando uma WAN 201**
- 13.5 Armazenamento e Encaminhamento (Store and Forward) 201**
- 13.6 Endereçamento Físico em uma WAN 202**
- 13.7 Encaminhamento Next-Hop 202**
- 13.8 Independência de Origem 204**



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

17.7	Arquitetura de Inter-rede	263
17.8	Obtendo Serviço Universal	264
17.9	Uma Rede Virtual	264
17.10	Protocolos para Ligação Inter-redes	264
17.11	Significado de Ligação Inter-redes e TCP/IP	265
17.12	Divisão em Camadas e Protocolos TCP/IP	265
17.13	Hosts, Roteadores e Camadas de Protocolo	267
17.14	Resumo	268

Capítulo 18 IP: Endereços de Protocolo Inter-rede 270

18.1	Introdução	271
18.2	Endereços para a Inter-rede Virtual	271
18.3	O Esquema de Endereçamento IP	272
18.4	A Hierarquia de Endereços IP	272
18.5	Classes Originais de Endereços IP	273
18.6	Computando a Classe de um Endereço	274
18.7	Notação Decimal Pontilhada	274
18.8	Classes e Notação Decimal Pontilhada	276
18.9	Divisão do Espaço de Endereçamento	276
18.10	Autoridade para Endereços	277
18.11	Um Exemplo de Endereçamento de Classes	277
18.12	Endereçamento Sub-rede e Endereçamento Sem Classes	277
18.13	Máscaras de Endereçamento	278
18.14	Notação CIDR	279
18.15	Um Exemplo de Bloco de Endereço CIDR	279
18.16	Endereços de Host CIDR	280
18.17	Endereços IP Especiais	281
18.18	Resumo de Endereços IP Especiais	282
18.19	A Forma de Endereço de Broadcast de Berkeley	283
18.20	Roteadores e o Princípio de Endereçamento IP	283
18.21	Hosts Multi-Homed	283
18.22	Resumo	284

Capítulo 19 Amarração (Binding) de Endereços de Protocolo 286

19.1	Introdução	287
19.2	Endereços de Protocolo e Entrega de Pacotes	287

19.3	Resolução de Endereços	288
19.4	Técnicas de Resolução de Endereços	289
19.5	Resolução de Endereços com Pesquisa de Tabela	289
19.6	Resolução de Endereço com Computação de Forma Fechada	290
19.7	Resolução de Endereços com Troca de Mensagens	291
19.8	Protocolo de Resolução de Endereços	292
19.9	Entrega de Mensagem ARP	293
19.10	Formato de Mensagem ARP	293
19.11	Enviando uma Mensagem ARP	295
19.12	Identificando Quadros ARP	295
19.13	Armazenando Respostas ARP na Cache	295
19.14	Processando uma Mensagem ARP que Chega	296
19.15	Divisão em Camadas, Resolução de Endereços, Endereços de Protocolo	297
19.16	Resumo	297

Capítulo 20 *Datagrama IP e Encaminhamento de Datagramas* 299

20.1	Introdução	301
20.2	Serviço sem Conexão	301
20.3	Pacotes Virtuais	301
20.4	O Datagrama IP	302
20.5	Encaminhando um Datagrama IP	303
20.6	Endereços IP e Entradas da Tabela de Roteamento	303
20.7	O Campo de Máscara e o Encaminhamento de Datagramas	305
20.8	Destino e Endereços de Próximo Hop	305
20.9	Entrega de Melhor Esforço (Best-Effort Delivery)	306
20.10	O Formato de Cabeçalho do Datagrama IP	306
20.11	Resumo	307

Capítulo 21 *Encapsulamento IP, Fragmentação e Remontagem* 309

21.1	Introdução	311
21.2	Transmissão de Datagrama e Quadros	311
21.3	Encapsulamento	311
21.4	Transmissão Através de uma Inter-Rede	312
21.5	MTU, Tamanho de Datagrama e Encapsulamento	313
21.6	Remontagem	315
21.7	Identificando um Datagrama	315

- 21.8 Perda de Fragmento 315
- 21.9 Fragmentando um Fragmento 316
- 21.10 Resumo 316

Capítulo 22 O Futuro IP (IPv6) 318

- [22.1 Introdução 319](#)
- [22.2 O Sucesso do IP 319](#)
- [22.3 A Motivação para Mudar 320](#)
- [22.4 Um Nome e um Número de Versão 320](#)
- [22.5 Características do IPv6 321](#)
- [22.6 Formato de Datagrama do IPv6 321](#)
- 22.7 Formato do Cabeçalho de Base de IPv6 322
- 22.8 Como o IPv6 Trata de Múltiplos Cabeçalhos 323
- [22.9 Fragmentação, Remontagem e MTU do Caminho 324](#)
- [22.10 O Propósito de Múltiplos Cabeçalhos 325](#)
- 22.11 Endereçamento do IPv6 326
- [22.12 Notação Hexadecimal de Dois pontos do IPv6 327](#)
- [22.13 Resumo 327](#)

Capítulo 23 Um Mecanismo de Relatório de Erro (ICMP) 329

- [23.1 Introdução 331](#)
- [23.2 Semântica do Melhor Esforço e Detecção de Erros 331](#)
- [23.3 Internet Control Message Protocol \(Protocolo de Mensagens de Controle Inter-rede\) 332](#)
- 23.4 Transporte de Mensagens do ICMP 334
- 23.5 Usando Mensagens de ICMP para Testar Alcançabilidade 334
- 23.6 Usando ICMP para Traçar uma Rota 335
- 23.7 O Último Endereço Impresso pelo Traceroute 335
- 23.8 Usando ICMP para Descoberta do MTU do Caminho 336
- 23.9 Resumo 337

Capítulo 24 UPD: Serviço de Transporte de Datagrama 338

- 24.1 Introdução 339
- 24.2 A Necessidade de Protocolos de Transporte Fim-a-Fim 339
- 24.3 O Protocolo do Usuário de Datagrama 340
- 24.4 O Paradigma sem Conexão 340

- 24.5** Interface Orientada à Mensagem 341
- 24.6** Semânticas de Comunicação UDP 341
- 24.7** Interação Arbitrária 342
- 24.8** Suporte para Unicast, Multicast e Broadcast 342
- 24.9** Identificação do Endpoint com Números de Portas de Protocolos 342
- 24.10** Formato de Datagrama de UDP 343
- 24.11** O Checksum do UDP e o Pseudo Cabeçalho 343
- 24.12** Encapsulamento do UDP 344
- 24.13** Resumo 344

Capítulo 25 TCP: Serviço de Transporte Confiável 346

- 25.1** Introdução 347
- 25.2** A Demanda por Transporte Confiável 347
- 25.3** O Transmission Control Protocol 348
- 25.4** O Serviço que o TCP Fornece para Aplicativos 348
- 25.5** Serviço Fim-a-Fim e Datagramas 349
- 25.6** Obtendo Confiabilidade 349
- 25.7** Perda de Pacote e Retransmissão 350
- 25.8** Retransmissão Adaptativa 351
- 25.9** Comparação de Tempos de Retransmissão 351
- 25.10** Buffers, Controle de Fluxo e Janelas 352
- 25.11** Three-Way Handshake 353
- 25.12** Controle de Congestionamento 354
- 25.13** Formato do Segmento de TCP 355
- 25.14** Resumo 356

Capítulo 26 Tradução de Endereços de Rede 358

- 26.1** Introdução 359
- 26.2** O Requerimento para Endereços Únicos 359
- 26.3** Tecnologia de Tradução de Endereços de Rede 360
- 26.4** Topologia de NAT 360
- 26.5** O Modelo Cliente-Servidor do DNS 360
- 26.6** Tradução de Endereços Básicos 361
- 26.7** Tabela de Tradução 361
- 26.8** Splicing de NAPT e TCP 362
- 26.9** Outras Variantes: Twice NAT e CAT 363

- 26.10 Software de NAT e Sistemas para Uso em Casa** 363
- 26.11 Resumo** 364

Capítulo 27 Roteamento na Internet 366

- 27.1 Introdução** 367
- 27.2 Roteamento Estático versus Roteamento Dinâmico** 367
- 27.3 Roteamento Estático em Hosts e uma Rota Padrão** 368
- 27.4 Roteamento Dinâmico e Roteadores** 369
- 27.5 Roteamento na Internet Global** 370
- 27.6 Conceito de Sistema Autônomo** 370
- 27.7 Os Dois Tipos de Protocolos de Roteamento** 371
- 27.8 Rotas e Tráfego de Dados** 373
- 27.9 O Border Gateway Protocol (BGP)** 373
- 27.10 O Routing Information Protocol (RIP)** 374
- 27.11 Formato dos Pacotes RIP** 375
- 27.12 O Open Shortest Path First Protocol (OSPF)** 376
- 27.13 Um Exemplo de Gráfico OSPF** 377
- 27.14 Áreas OSPF** 377
- 27.15 Roteamento Multicast** 378
- 27.16 Resumo** 381

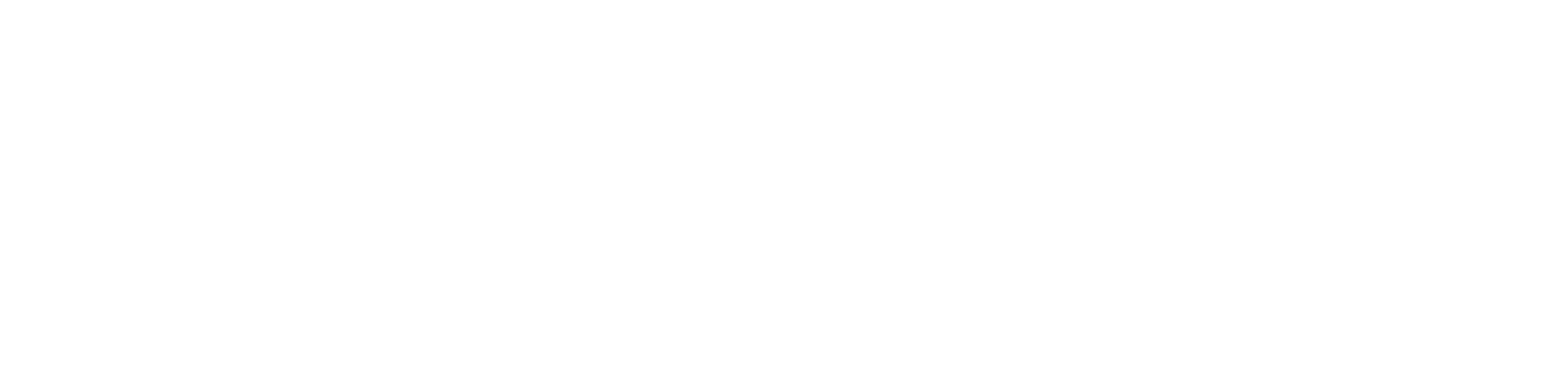
Aplicativos de Rede

Capítulo 28 Interação Cliente-Servidor 383

- 28.1 Introdução** 385
- 28.2 A Funcionalidade que o Software aplicativo Fornece** 386
- 28.3 A Funcionalidade que uma Inter-rede Fornece** 386
- 28.4 Estabelecendo Contato** 386
- 28.5 O Paradigma Cliente-Servidor** 387
- 28.6 Características de Clientes e Servidores** 387
- 28.7 Programas Servidores e Computadores da Classe Servidor** 388
- 28.8 Requisições, Respostas e Direção do Fluxo de Dados** 388
- 28.9 Protocolos de Transporte e Interação Cliente-Servidor** 388
- 28.10 Múltiplos Serviços em um Computador** 389
- 28.11 Identificando um Serviço Particular** 390
- 28.12 Múltiplas Cópias de um Servidor para um Único Serviço** 390



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

- 32.15** Acesso à Caixa de Correio 447
- 32.16** Conexões de Dialup e POP 448
- 32.17** Resumo 449

Capítulo 33 Telefonia IP (VoIP) 451

- 33.1** Introdução 453
- 33.2** A Motivação e o Desafio da Telefonia IP 453
- 33.3** Codificação, Transmissão e Playback 454
- 33.4** Protocolos e Sistemas de Sinalização (Signaling) 454
- 33.5** Um Sistema de Telefonia IP Básico 455
- 33.6** Interoperação com Outros Sistemas Telefônicos 456
- 33.7** Terminologia Alternativa e Conceitos 456
- 33.10** Camadas H.323 459
- 33.11** Características da SIP e Identificação do Usuário 459
- 33.12** Métodos SIP 460
- 33.13** Um Exemplo de Sessão SIP 461
- 33.14** Mapeamento e Roteamento de Números de Telefone 462
- 33.15** Telefones IP e Energia Elétrica 462
- 33.16** Resumo 462

Capítulo 34 Transferência de Arquivos e Acesso a Arquivos Remotos 464

- 34.1** Introdução 465
- 34.2** Transferência de Arquivos Generalizada 465
- 34.3** O Protocolo de Transferência de Arquivos 466
- 34.4** Modelo de FTP e Interface com o Usuário Geral 466
- 34.5** Comandos do FTP 466
- 34.6** Conexões, Autorização e Permissões de Arquivo 467
- 34.7** Acesso Anônimo a Arquivos 468
- 34.8** Transferência de Arquivos em Uma ou Outra Direção 468
- 34.9** Expansão de Coringa (Wildcard) em Nomes de Arquivo 469
- 34.10** Tradução de Nome de Arquivo 469
- 34.11** Diretórios Variáveis e Listagem de Conteúdo 470
- 34.12** Tipos de Arquivos e Modos de Transferência 470
- 34.13** Exemplo de Uso de FTP 471
- 34.14** Saída Verbosa (Verbose Output) 473
- 34.15** Interação Cliente-Servidor em FTP 473

34.16	Controle e Conexões de Dados	473
34.17	Conexões de Dados e Fim de Arquivo	474
34.18	Trivial File Transfer Protocol	474
34.19	Sistema de Arquivos de Rede	475
34.20	Resumo	476

Capítulo 35 Páginas da World Wide Web e Navegação 478

35.1	Introdução	481
35.2	Interface do Navegador	481
35.3	Hipertexto e Hipermídia	481
35.4	Representação de Documento	482
35.5	Formato HTML e Representação	482
35.6	Exemplos de Etiquetas HTML de Formatação	483
35.7	Cabeçalhos	484
35.8	Listas	484
35.9	Embutindo Imagens Gráficas em uma Página Web	485
35.10	Identificando uma Página	485
35.11	Links de Hipertexto de um Documento para Outro	486
35.12	Interação Cliente-Servidor	487
35.13	Transporte de Documentos Web e HTTP	488
35.14	Arquitetura do Navegador	488
35.15	Clientes Opcionais	490
35.16	Caching em Navegadores Web	491
35.17	Suporte HTTP para Caching	491
35.18	Protocolos de Transferência Alternativos	492
35.19	Outras Linguagens Markup	492
35.20	Resumo	492

Capítulo 36 Tecnologias para Documentos Web Dinâmicos (CGI, ASP, JSP, PHP e Coldfusion) 493

36.1	Introdução	495
36.2	Três Tipos Básicos de Documentos Web	495
36.3	Vantagens e Desvantagens de Cada Tipo de Documento	496
36.4	Implementação de Documentos Dinâmicos	497
36.5	O Padrão CGI	497
36.6	Saída de um Programa CGI	497

36.7	Um Programa CGI Exemplo	498
36.8	Parâmetros e Variáveis de Ambiente	500
36.9	Informações de Estado e Cookies	500
36.10	Um Script CGI com Informações de Estado a Longo Prazo	501
36.11	Um Script CGI com Informações de Estado a Curto Prazo	502
36.12	Formulários e Interação	503
36.13	As Tecnologias de Script do Lado do Servidor	503
36.14	Resumo	506

Capítulo 37 *Tecnologia para Documentos Web Ativos (Java e JavaScript)* 508

37.1	Introdução	509
37.2	Atualização Contínua com Server Push e Client Pull	509
37.3	Documentos Ativos e Sobrecarga do Servidor	510
37.4	Representação de Documento e Tradução Ativas	510
37.5	Tecnologia Java	511
37.6	A Linguagem de Programação de Java	511
37.7	O Ambiente de Execução (Runtime) do Java	512
37.8	A Biblioteca Java	513
37.9	Um Conjunto de Ferramentas Gráficas	514
37.10	Usando Java Gráfico em um Computador em Particular	515
37.11	Interpretadores e Navegadores Java	516
37.12	Compilando um Programa Java	516
37.13	Um Applet Exemplo	517
37.14	Invocando um Applet	518
37.15	Exemplo de Interação com um Navegador	519
37.16	Erros e Tratamento de Exceções	520
37.17	Tecnologia de JavaScript	521
37.18	Alternativas	522
37.19	Resumo	523

Capítulo 38 *RPC e Middleware* 525

38.1	Introdução	527
38.2	Programando Clientes e Servidores	529
38.3	Paradigma de Chamada Remota de Procedimento	530
38.4	Paradigma RPC	531

38.5 Stubs de Comunicação 532

- 38.6 Representação de Dados Externos 533**
- 38.7 Middleware e Middleware Orientado a Objetos 534**
- 38.8 Resumo 537**

Capítulo 39 Gerência de Redes (SNMP) 537

- 39.1 Introdução 539**
- 39.2 Administrando uma Inter-Rede 539**
- 39.3 O Perigo de Falhas Escondidas 539**
- 39.4 Software de Gerência de Rede 540**
- 39.5 Clientes, Servidores, Gerentes e Agentes 540**
- 39.6 Simple Network Management Protocol 541**
- 39.7 Paradigma de Carga e Armazenamento (Fetch-Store) 541**
- 39.8 A MIB e Nomes de Objetos 542**
- 39.9 A Variedade de Variáveis MIB 542**
- 39.10 Variáveis MIB que Correspondem a Arrays 543**
- 39.11 Resumo 543**

Capítulo 40 Segurança de Rede 545

- 40.1 Introdução 547**
- 40.2 Políticas e Redes Seguras 547**
- 40.3 Aspectos de Segurança 547**
- 40.4 Responsabilidade e Controle 548**
- 40.5 Mecanismos de Integridade 548**
- 40.6 Controle de Acesso e Senhas 549**
- 40.7 Criptografia e Privacidade 549**
- 40.8 Criptografia de Chave Pública 550**
- 40.9 Autenticação com Assinaturas Digitais 550**
- 40.10 Conceito de Firewall de Internet 551**
- 40.11 Filtragem de Pacotes Usando Portas 552**
- 40.12 Usando Filtros de Pacotes para Criar uma Firewall 553**
- 40.13 Redes Privadas Virtuais 554**
- 40.14 Tunelamento 555**
- 40.15 Tecnologias de Segurança 556**
- 40.16 Resumo 557**

Capítulo 41 Inicialização (Configuração) 559

- 41.1 Introdução 561**
- 41.2 Software de Protocolo Bootstrapping 561**
- 41.3 Parâmetros de Protocolo 562**
- 41.4 Configuração de Protocolo 562**
- 41.5 Exemplos de Itens que Precisam ser Configurados 562**
- 41.6 Configuração a partir de Armazenamento Estável 563**
- 41.7 A Necessidade de Automatizar a Configuração de Protocolo 563**
- 41.8 Métodos para Configuração Automatizada de Protocolo 563**
- 41.9 O Endereço Usado para Encontrar um Endereço 564**
- 41.10 Uma Seqüência de Protocolos Usados Durante o Bootstrap 565**
- 41.11 Protocolo Bootstrap (BOOTP) 565**
- 41.12 Dynamic Host Configuration Protocol (DHCP) 567**
- 41.13 Otimizações em DHCP 568**
- 41.14 Acesso Indireto ao Servidor Através de um Relay 568**
- 41.15 Formato da Mensagem de DHCP 568**
- 41.16 DHCP e Nomes de Domínio 569**
- 41.17 Resumo 570**

Apêndice 1 Glossário de Termos e Abreviações de Ligação em Redes 573

Apêndice 2 O Conjunto de Caracteres ASCII 609

Apêndice 3 Máscara de Endereço em Decimal com Pontos 611

Apêndice 4 Como Usar o CD-ROM Incluído Neste Livro 613

Bibliografia 617

Índice 625



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

Introdução

1.1 Crescimento das redes de computadores

As redes de computadores têm crescido explosivamente. Há duas décadas, poucas pessoas tinham acesso a uma rede. Agora, a comunicação via computador transformou-se em uma parte essencial da infra-estrutura de todos. A ligação de computadores em rede é usada em cada aspecto dos negócios, incluindo propaganda, produção, transporte, planejamento, faturamento e contabilidade. Consequentemente, a maioria das corporações tem múltiplas redes. As instituições de ensino, em todos os níveis, do fundamental ao pós-graduação, estão utilizando redes de computadores para fornecer a estudantes e professores acesso instantâneo a informações em bibliotecas on-line em todo o mundo. Escritórios governamentais em níveis federal, estadual e municipal utilizam redes, assim como organizações militares. Em resumo, as redes de computadores estão em toda parte.

O crescimento contínuo da Internet global é um dos fenômenos mais interessantes e excitantes em redes. Há pouco mais de vinte anos, a Internet era um projeto de pesquisa que envolvia algumas dúzias de sites. Hoje, ela cresceu e se tornou um sistema de comunicação produtivo que alcança milhões de pessoas em todos os países do mundo. Nos Estados Unidos, a Internet conecta a maioria das corporações, faculdades e universidades, assim como escritórios federais, estaduais e municipais. Também alcança a maioria das escolas. Além disso, muitas residências têm acesso à Internet através das conexões de linha discada, cable-modems, DSL e tecnologias sem fio. Uma prova do impacto da Internet na sociedade pode ser percebida nas propagandas em revistas e na televisão, que freqüentemente contêm referências a sites da Web que fornecem informações adicionais sobre os produtos e serviços dos anunciantes.

O crescimento das ligações de computadores em rede tem também um impacto econômico. As redes de dados têm disponibilizado o telecommuting aos indivíduos e mudaram a comunicação no mundo dos negócios. Além disso, uma indústria inteira surgiu e desenvolve produtos, serviços e tecnologias de rede. A popularidade e a importância das redes de computadores têm produzido uma forte demanda em todos os empregos para pessoas com maior conhecimento sobre o assunto. As empresas necessitam de empregados para planejar, adquirir, instalar, operar e gerenciar os sistemas de hardware e software que fazem redes de computadores e inter-redes. Além disso, a programação computacional não é mais restrita a computadores individuais; espera-se que os programadores projetem e executem software aplicativo que possa se comunicar com o software em outros computadores.

1.2 Complexidade em sistemas de rede

A ligação de computadores em rede é um assunto complexo. Existem muitas tecnologias, e cada uma possui características que a distingue das outras. Muitas organizações criaram, independentemente, padrões de ligação em rede que não são totalmente compatíveis. Muitas empresas criaram produtos comerciais e serviços de ligação em rede que usam as tecnologias de maneiras não-convencionais. Finalmente, a ligação em rede é complexa porque existem múltiplas tecnologias que podem ser usadas para interconectar duas ou mais redes. Como consequência, são possíveis muitas combinações de redes.

A ligação de computadores em rede pode ser especialmente confusa para um novato porque não há nenhuma teoria de base que explique o relacionamento entre todas as partes. Várias organizações e grupos de pesquisa têm tentado definir os modelos conceituais que podem ser usados para explicar as diferenças e similaridades entre hardware de rede e sistemas de software. Infelizmente, o conjunto de tecnologias é diverso e está mudando rapidamente; os modelos são tão simples que não distinguem os detalhes ou tão complexos que não ajudam a facilitar o assunto.

A falta de uma teoria de embasamento produziu um outro desafio para os novatos: não há uma terminologia simples e uniforme para os conceitos sobre ligação de computadores em rede. Uma vez que várias organizações definem tecnologias e padrões sobre ligação em rede, existem vários termos para um dado conceito. Os profissionais freqüentemente usam um termo técnico de uma tecnologia ao se referirem a uma característica análoga de outra tecnologia. E os termos técnicos são confundidos às vezes com nomes de produtos populares. Assim, além de um grande conjunto de termos e de siglas que contêm muitos sinônimos, o jargão da ligação em rede contém termos que são freqüentemente abreviados, mal-empregados ou associados a produtos.

1.3 Dominando a complexidade

Para dominar a complexidade, deve-se olhar para além dos detalhes e concentrar-se em entender os conceitos. Por exemplo, embora não seja importante compreender os detalhes sobre os fios usados para conectar computadores a uma rede específica, é importante entender as poucas categorias básicas de esquemas de fiação e as vantagens de cada um. Similarmente, embora não seja importante aprender os detalhes de como um protocolo de comunicação particular trata uma rede congestionada, é importante saber o que é congestionamento e porque ele deve ser tratado.

1.4 Conceitos e terminologia

Este livro foi escrito para ajudar a superar a complexidade, concentrando-se em conceitos e evitando detalhes desnecessários. Explica a finalidade de cada tecnologia de ligação em rede, apresenta suas vantagens e desvantagens e descreve algumas das consequências ao usá-las. Sempre que possível, são utilizadas analogias e ilustrações para simplificar as explicações.

Além de cobrir os conceitos e as tecnologias, o livro apresenta terminologia sobre ligação de computadores em rede. Quando é apresentado um conceito novo, sua terminologia é definida. O livro também mostra as abreviaturas e os sinônimos populares utilizados pelos profissionais. A terminologia está resumida em um glossário no Apêndice 1, que serve como uma referência rápida para os muitos termos e siglas definidas ao longo do livro.

1.5 O valor da experiência prática

Este texto fornece uma visão abrangente do material essencial para o iniciante; conhecimento aprofundado resulta somente da experiência pessoal. Portanto, os leitores são fortemente recomendados a ganhar tanta experiência prática quanto for possível com softwares e hardwares de redes.

Possibilidades incluem: construir aplicativos que se comuniquem pela rede, configurar sistemas de rede, observar os protocolos em ação e medir a performance do sistema. O texto para acompanhamento, *Hands-On Networking** , contém muitas sugestões de experiências e projetos. Alguns dos exercícios do livro se referem ao *Hands-On Networking* e recomendam experimentos específicos que ajudarão o leitor a ter uma compreensão aprofundada do material.

1.6 Organização do livro

O texto começa com capítulos que introduzem aplicativos e programação de redes. Leitores que têm acesso a um computador são encorajados a construir e usar aplicativos que utilizem a Internet enquanto prosseguem com a leitura. As quatro partes do livro que seguem os capítulos sobre aplicativos explicam como as tecnologias subjacentes funcionam. A primeira parte descreve a transmissão de dados. Ela explica que, no nível mais baixo, são usados sinais elétricos que viajam através dos fios para carregar informações e mostra como os dados podem ser codificados usando sinais elétricos. Os capítulos da primeira parte não fornecem detalhes para engenheiros que projetam hardware de ligação em rede. Em vez disso, fornecem descrições gerais dos princípios e das realidades práticas da transmissão de dados e de suas consequências para as redes de computadores.

A segunda parte do livro se concentra na transmissão de pacotes. Explica porque as redes de computadores usam pacotes e mostra como são agrupados dados em pacotes para transmissão. Essa seção introduz as duas categorias básicas de redes de computadores: Redes Locais (*Local Area Networks, LANs*) e Redes de Longo Alcance (*Wide Area Networks, WANs*). Explica as diferenças entre as duas e revisa exemplos de tecnologias. Finalmente, discute os conceitos importantes de endereçamento e roteamento, explicando como uma rede roteia um pacote a seu destino.

A terceira parte do livro cobre a ligação inter-redes – a idéia importante que permite que as tecnologias heterogêneas de rede sejam combinadas em um grande sistema integrado de comunicação. O livro explica TCP/IP, a tecnologia do protocolo usada na Internet global.

A quarta parte do livro retorna às aplicações da ligação de computadores em rede. Os capítulos começam explicando o modelo cliente-servidor de interação. Capítulos posteriores usam o modelo para explicar como programas aplicativos fornecem serviços como correio eletrônico e navegação na Web.

1.7 Resumo

O grande conjunto de tecnologias, produtos e esquemas de interconexão torna a ligação em rede um assunto complexo. Muitas organizações têm definido padrões competitivos, e a maioria das redes incorpora componentes que usam padrões múltiplos. Além disso, não existe uma teoria que possa ser usada para explicar como as partes se encaixam. Consequentemente, a terminologia e o jargão usados na ligação em rede são complexos e confusos. Para dominar tal complexidade, é importante enfatizar a compreensão dos conceitos e da terminologia.

A obra, que enfatiza a conceituação, cobre todos os aspectos da ligação de computadores em redes: desde as aplicações comuns até os detalhes de baixo nível sobre ligações e sinais. Explica como as tecnologias são usadas para formar a Internet. O livro de acompanhamento, *Hands-On Networking*, contém um conjunto de projetos que examinam as tecnologias de rede disponíveis e fornecem experiência prática ao leitor.

* Nota: Para mais informações sobre esse material, acessar o endereço www.netbooks.cs.purdue.edu

Sumário do Capítulo

- | | | |
|------------|------------------------------------|----|
| 2.1 | Introdução | 37 |
| 2.2 | Compartilhamento de Recursos | 37 |
| 2.3 | Crescimento da Internet | 38 |
| 2.4 | Testando a Internet | 39 |
| 2.5 | Interpretando uma Resposta do Ping | 41 |
| 2.6 | Traçando uma Rota | 43 |
| 2.7 | Resumo | 44 |

Motivação e Ferramentas

2.1 Introdução

Antes de examinar tecnologias subjacentes da rede, será instrutivo considerar a motivação para a ligação em redes e ligação inter-redes e examinar alguns dos serviços que tais sistemas fornecem. Além de rever algumas das motivações iniciais, este capítulo discute o tamanho e o crescimento rápido da Internet e introduz algumas ferramentas básicas que podem ser usadas para explorar as redes.

2.2 Compartilhamento de recursos

Algumas das primeiras redes de computadores foram construídas para expandir equipamentos de computação já existentes. Por exemplo, foram projetadas redes que permitiam a múltiplos computadores acessar um dispositivo periférico compartilhado, como uma impressora ou um disco. Isto é, ao invés de conectar um dispositivo periférico a um único computador, o dispositivo era conectado a uma rede, o que permitia acesso a partir de qualquer computador conectado à rede. As motivações iniciais para redes de dados de larga escala não surgiram do desejo de compartilhar dispositivos periféricos ou mesmo de fornecer uma outra forma das pessoas se comunicarem. Em vez disso, as primeiras redes foram projetadas para compartilhar poder computacional em grande escala.

Para entender o problema, é importante saber que os primeiros computadores digitais eram extremamente caros e escassos. Com a evolução da tecnologia, surgiram computadores com maior poder computacional e maior capacidade de armazenamento. O governo dos Estados Unidos, que financia grande parte da pesquisa em ciência e em engenharia, percebeu que os computadores eram cruciais aos avanços na ciência e na tecnologia. Uma vez que os computadores eram usados para analisar os dados dos experimentos, os programas funcionavam freqüentemente por horas, ou mesmo dias. O orçamento do governo para pesquisa era insuficiente para fornecer computadores a todos os cientistas e engenheiros.

A Agência de Projetos de Pesquisa Avançada (*Advanced Research Projects Agency, ARPA*), do Departamento de Defesa dos EUA, estava especialmente preocupada com a falta de computadores de alta potência. Muitos dos projetos de pesquisa da ARPA necessitavam de acesso a equipamen-



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

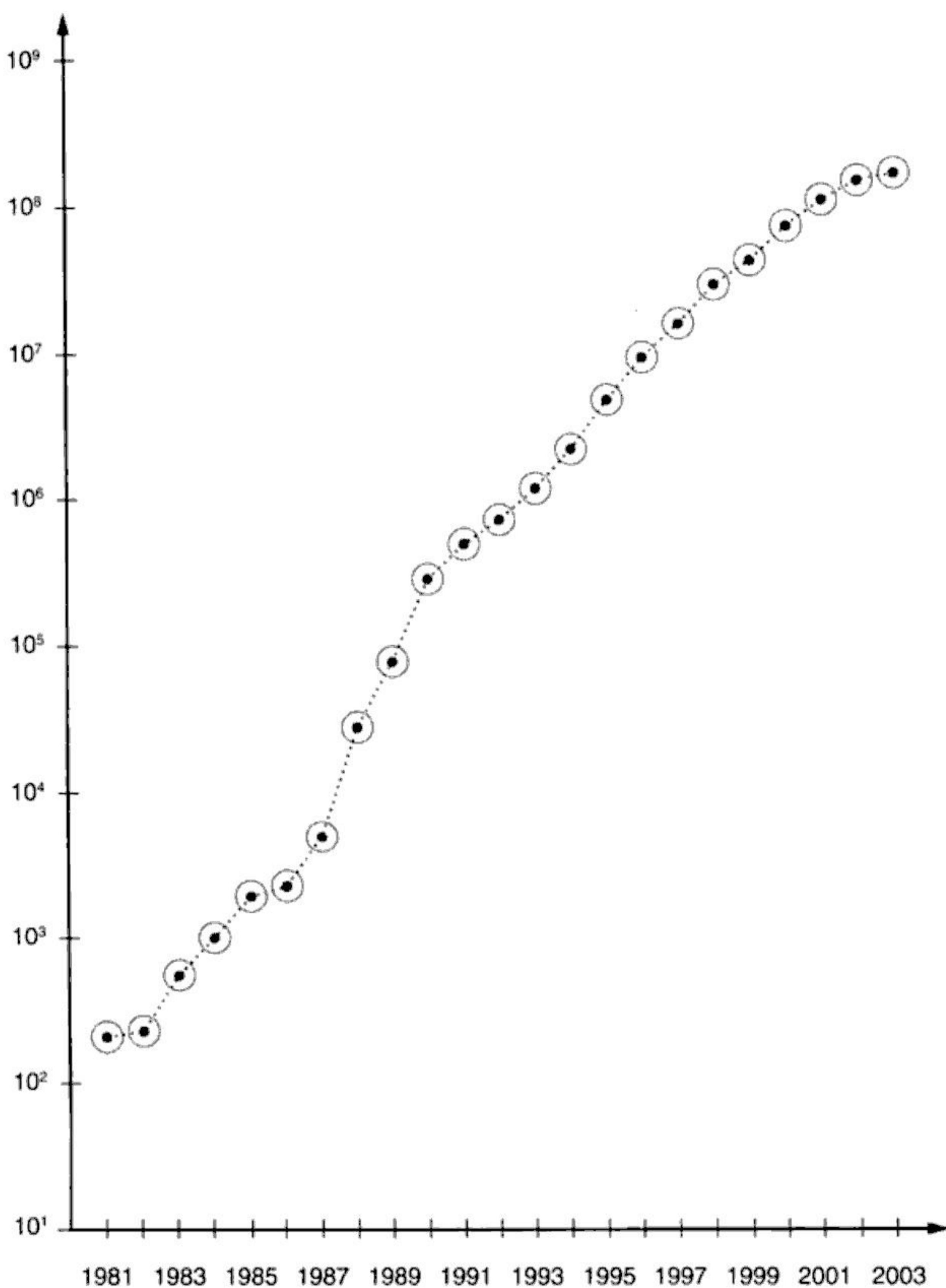


Figura 2.2 O crescimento da Internet de 1981 a 2003, plotado em uma escala logarítmica, ilustra o crescimento exponencial.

de Nome de Domínio) – o sistema que guarda nomes de computadores, como www.yahoo.com, junto com o endereço do computador – e então usa um programa que testa se o computador está atualmente on-line. Ferramentas usadas para testar a Internet estão disponíveis também para usuários.

Uma das ferramentas de sondagem mais simples consiste em um programa conhecido como *ping*². Quando um usuário invoca o ping, deve especificar um argumento que forneça o nome ou o endereço numérico de um computador remoto. Por exemplo, um usuário poderia usar o ping com um argumento que especificasse o computador www.netbook.cs.purdue.edu:

```
ping www.netbook.cs.purdue.edu
```

O programa ping envia uma mensagem ao computador especificado e espera uma resposta por um curto tempo. Se chega uma resposta, o ping relata ao usuário que o computador está vivo; caso contrário, relata que o computador não está respondendo. Por exemplo, a seguinte saída pode aparecer:

```
www.netbook.cs.purdue.edu is alive
```

Algumas versões do ping fornecem opções que permitem que o usuário especifique o tamanho do pacote enviado, que faça o ping computar o tempo de ida e volta (o tempo entre enviar uma mensagem e receber uma resposta) ou enviar repetidamente uma mensagem por segundo até que o programa seja parado. A Figura 2.3 mostra um exemplo da saída do ping quando as opções de cálculo de tempo e repetição estão ativas. O exemplo foi executado da estação de trabalho do autor ao destino www.sears.com.

2.5 Interpretando uma resposta do ping

Na Figura 2.3, o ping envia uma requisição a cada segundo e produz uma linha de saída para cada resposta recebida. A saída diz o tamanho do pacote recebido, o número de seqüência e o tempo de ida e volta em milissegundos. Quando o usuário interrompe o programa, o ping produz um resumo que especifica o número de pacotes enviados e recebidos, perda de pacotes e os tempos de ida e volta mínimo, médio e máximo.

A saída na figura mostra outra característica interessante. Embora o autor tenha especificado www.sears.com como o computador destino, o ping lista o nome de computador como *sears.com*. O Capítulo 31 discute nomes de computadores em detalhe. Por enquanto é suficiente saber que www.sears.com é meramente um pseudônimo para o computador *sears.com*.

A saída na Figura 2.3 mostra um tempo de ida e volta médio de 10 milissegundos, típico quando a Internet não está congestionada. Como comparação, a Figura 2.4 mostra o tempo de ida e volta da

```
PING www.sears.com: 56 data bytes
 64 bytes from 129.33.131.220: icmp_seq=0. time=10. ms
 64 bytes from 129.33.131.220: icmp_seq=1. time=11. ms
 64 bytes from 129.33.131.220: icmp_seq=2. time=11. ms
 64 bytes from 129.33.131.220: icmp_seq=3. time=10. ms
 64 bytes from 129.33.131.220: icmp_seq=4. time=10. ms
 -----www.sears.com PING Statistics-----
 5 packets transmitted, 5 packets received, 0% packet loss
 round-trip (ms) min/avg/max = 10/10/11
```

Figura 2.3 Exemplo de saída de um programa ping executado na estação de trabalho do autor. O destino foi www.sears.com, e o programa foi interrompido manualmente após o recebimento de cinco respostas.

² O Capítulo 23 explica os detalhes exatos e os protocolos usados pelo programa ping e outras ferramentas descritas neste capítulo.

```
PING Berkeley.EDU: 56 data bytes
64 bytes from chaparral.Berkeley.EDU (128.32.25.19): icmp_seq=0. time=49. ms
64 bytes from chaparral.Berkeley.EDU (128.32.25.19): icmp_seq=1. time=48. ms
64 bytes from chaparral.Berkeley.EDU (128.32.25.19): icmp_seq=2. time=48. ms
64 bytes from chaparral.Berkeley.EDU (128.32.25.19): icmp_seq=3. time=48. ms
64 bytes from chaparral.Berkeley.EDU (128.32.25.19): icmp_seq=4. time=48. ms
----Berkeley.EDU PING Statistics----
5 packets transmitted, 5 packets received, 0% packet loss
round-trip (ms) min/avg/max = 48/48/49
```

Figura 2.4 Exemplo de saída de um programa ping para o destino *berkeley.edu*, localizado na costa oeste. Os tempos de ida e volta são maiores do que os relatados na Figura 2.3.

estaçāo de trabalho do autor para um site na costa ocidental (a University of California, em Berkeley), e a Figura 2.5 mostra o tempo de ida e volta para um site na costa leste (o MIT, em Cambridge, Massachusetts), medidos no mesmo dia e aproximadamente na mesma hora. O caminho físcico até o MIT é mais curto do que o caminho físcico até Berkeley, o que é refletido na média dos tempos de ida e volta.

Pode parecer que o programa ping é demasiado simples para ser útil. Mesmo com as opções ativas, os tempos de ida e volta fornecem poucas informações ao usuário médio. Por exemplo, o ping não pode explicar porque o tempo necessário para alcançar o MIT é mais elevado do que o tempo médio para alcançar outras posições distantes. Mais importante, parece que o ping tem pouco a oferecer como uma ferramenta para depurar falhas de rede, porque a saída é apresentada somente quando um computador responde com sucesso. Quando nenhuma resposta é recebida, o ping não pode ajudar a determinar a razão. O computador remoto poderia ter sido desligado, desconectado da rede, sua interface de rede poderia ter falhado ou o software em execução poderia não estar respondendo ao ping. O computador local poderia estar desconectado da rede, a rede em que o computador remoto está conectado poderia ter falhado, ou o problema poderia ter sido causado pela falha de um computador ou rede intermediários. Finalmente, o ping às vezes falha porque a rede está tão congestionada com tráfego que os atrasos são longos demais. O ping não tem como determinar a causa do problema.

Outra razão pela qual o ping pode falhar para gerar uma resposta é menos sutil: algumas companhias configuraram seus sites para rejeitar pacotes de ping. A motivação para desativar o ping é a segurança – se uma corporação permite a entrada de tráfego de ping em seu site, este se torna suscetível a um ataque de *recusa de serviço (denial-of-service)* ou *sobrecarga (flooding)*, no qual chega uma quantidade tão grande de pacotes que os computadores e as redes da companhia não podem responder a pedidos legítimos. Para evitar tais ataques, a companhia meramente rejeita pacotes de ping antes de eles entrarem.

```
PING MIT.EDU: 56 data bytes
64 bytes from SICARIUS-SPATULATOS/MIT.EDU (18.7.21.77): icmp_seq=0. time=32. ms
64 bytes from SICARIUS-SPATULATOS/MIT.EDU (18.7.21.77): icmp_seq=1. time=31. ms
64 bytes from SICARIUS-SPATULATOS/MIT.EDU (18.7.21.77): icmp_seq=2. time=31. ms
64 bytes from SICARIUS-SPATULATOS/MIT.EDU (18.7.21.77): icmp_seq=3. time=31. ms
64 bytes from SICARIUS-SPATULATOS/MIT.EDU (18.7.21.77): icmp_seq=4. time=31. ms
----MIT.EDU PING Statistics----
5 packets transmitted, 5 packets received, 0% packet loss
round-trip (ms) min/avg/max = 31/31/32
```

Figura 2.5 Exemplo de saída de um programa ping para o destino *mit.edu*, que está localizado na costa leste dos Estados Unidos. Os tempos de ida e volta são significativamente maiores do que nos exemplos anteriores.

Você pode ficar surpreso ao aprender que, apesar de suas limitações, o ping é bastante usado como uma ferramenta de diagnóstico. De fato, os administradores de rede freqüentemente rodam o ping assim que descobrem uma falha para determinar quais partes da rede estão operando corretamente e quais falharam. Os resultados os ajudam a localizar rapidamente a falha.

2.6 Traçando uma rota

Os administradores da rede usam outra ferramenta, *traceroute*, para determinar os computadores intermediários ao longo do trajeto até um destino remoto. Como o ping, o traceroute aceita um argumento que especifica um nome de computador remoto ou endereço. Por exemplo, o seguinte comando seguirá um trajeto do computador do usuário a www.netbook.cs.purdue.edu:

```
traceroute www.netbook.cs.purdue.edu
```

O traceroute determina as rotas intermediárias ao longo de um trajeto até o destino³ e imprime uma linha para cada um. A Figura 2.6 mostra a saída do traceroute para o destino mit.edu.

O traceroute fornece mais informações do que o ping. Por exemplo, a figura mostra doze linhas da saída para o trajeto entre a estação de trabalho do autor e o computador destino em Berkeley. Uma linha corresponde a cada um de onze computadores intermediários, e uma corresponde ao destino final propriamente dito. No jargão de redes, dizemos que o destino está doze hops distante da fonte. É interessante notar que ele não pode ser utilizado para todos os destinos porque alguns administradores de redes preferem desativá-lo, para prevenir que invasores obtenham informações detalhadas sobre sua arquitetura.

A linha treze da Figura 2.6 ilustra outras características do traceroute: um relatório de pacote perdido. Traceroute envia três sinais para cada computador intermediário. Quando as três respostas chegam, imprime o nome do computador intermediário e dá o tempo de ida e volta mínimo, médio e máximo. A linha treze começa com dois asteriscos porque dois dos três sinais não receberam resposta (p. ex., pacotes foram perdidos). Em outro teste, todos os sinais foram recebidos corretamente. Então pode-se concluir que a perda foi uma condição temporária, provavelmente causada pelo congestionamento de um dos caminhos entre a fonte e o destino.

```
traceroute to berkeley.edu (128.32.25.19), 30 hops max, 40 byte packets
1 cisco5 (128.10.2.250) 0.958 ms 0.746 ms 0.705 ms
2 cisco-tel3-242.tcom.purdue.edu (128.210.242.24) 552.223 ms 443.787 ms 1.206 ms
3 tel-210-m10-01-242.tcom.purdue.edu (128.210.242.251) 0.736 ms 1.123 ms 0.924 ms
4 gigapop.tcom.purdue.edu (192.5.40.14) 16.793 ms 2.789 ms 2.614 ms
5 abilene-ul.indiana.gigapop.net (192.12.206.249) 2.702 ms 11.644 ms 2.521 ms
6 kscyng-iplsng.abilene.ucaid.edu (198.32.8.81) 12.938 ms 12.088 ms 21.926 ms
7 snvng-kscyng.abilene.ucaid.edu (198.32.8.102) 47.417 ms 47.088 ms 47.137 ms
8 198.32.249.161 (198.32.249.161) 47.435 ms 47.191 ms 47.093 ms
9 BERK-SUNV.POS.calren2.net (198.32.249.13) 48.464 ms 48.265 ms 48.445 ms
10 pos1-0.inr-000-eva.Berkeley.EDU (128.32.0.89) 48.223 ms 48.434 ms 48.664 ms
11 vlan199.inr-202-doecev.Berkeley.EDU (128.32.0.203) 49.085 ms 49.358 ms 48.877 ms
12 vlan210.inr-203-eva.Berkeley.EDU (128.32.255.10) 48.660 ms 48.933 ms 48.848 ms
13 * * chaparral.Berkeley.EDU (128.32.25.19) 48.794 ms
```

Figura 2.6 Exemplo de saída da execução do traceroute na estação de trabalho do autor para o destino berkeley.edu, o qual é um alias para o computador chaparral.Berkeley.edu.

³ O Capítulo 17 explica o propósito de roteadores.

2.7 Resumo

A Agência de Projetos de Pesquisa Avançada (ARPA) financiou muitas das pesquisas iniciais em redes como uma maneira de compartilhar recursos computacionais entre seus pesquisadores. Mais tarde, mudou seu foco para ligação inter-redes e começou a Internet, que tem crescido exponencialmente há muitos anos.

Algumas das ferramentas usadas para testar a Internet estão disponíveis para usuários. O programa ping envia uma mensagem a um computador remoto e relata se o computador responde; o programa traceroute identifica computadores intermediários ao longo de um trajeto até um destino remoto.

Softwares de ping e traceroute estão incluídos em muitos sistemas operacionais; código fonte para versões mais avançadas também está⁴. Por exemplo, é possível encontrar versões de código fonte em:

<http://www.shareware.com/>

Para acessar o traceroute através da Web, contate:

<http://www.net.cmu.edu/cgi-bin/netops.cgi>

Exercícios

- 2.1 Use o programa ping para testar se você pode alcançar computadores em sua rede local (experimento 3.4 no livro *Hands-On Networking*).
- 2.2 Se a sua versão do ping reporta o tempo necessário para obter uma resposta, experimente descobrir se os atrasos de rede variam durante o dia.
- 2.3 Use o ping para medir os tempos de ida e volta aos destinos na Internet (por exemplo, aos sites da Web). Qual é o tempo máximo que você encontra?
- 2.4 Faça experiências com a opção tamanho do pacote no ping. Como o tamanho do pacote afeta o tempo de ida e volta?
- 2.5 Compare a saída do programa ping para um computador que esteja desligado com a saída do ping para um endereço não-existente (por exemplo, 10.0.0.50). Elas diferem?
- 2.6 Use o programa traceroute para encontrar o número de hops entre seu computador e destinos remotos (por exemplo, a sites bem conhecidos da Web). Qual é o número máximo de hops que você pode encontrar?
- 2.7 Compare os tempos de ida e volta relatados pelo ping com o número de hops relatados pelo traceroute para um conjunto de destinos. Há uma correlação entre um atraso mais longo e uma contagem de hops mais elevada?
- 2.8 A tecnologia da Internet está documentada em uma série de relatórios conhecida como *Request For Comments (RFC)*. A RFC 2151, que pode ser encontrada no CD-ROM que acompanha este texto, descreve ferramentas disponíveis na Internet. Quais ferramentas o documento RFC descreve que não são descritas neste capítulo?
- 2.9 Encontre duas outras ferramentas de rede que são projetadas para permitir ao administrador medir, diagnosticar ou depurar uma rede.
- 2.10 Use uma ferramenta de captura de pacotes para obter um exemplo de pacote na sua rede local (experimento 3.5 em *Hands-On Networking*).

⁴ O sistema Windows utiliza o nome tracert em vez de traceroute.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

Aplicativos e Programação em Rede

3.1 Introdução

O capítulo anterior menciona alguns dos serviços que a rede fornece, discute o tamanho e o crescimento da Internet e explica algumas ferramentas de medição. Embora este texto não seja sobre programação, este capítulo descreve redes de computadores do ponto de vista dos programadores. Depois de esboçar brevemente as facilidades disponíveis da rede para os programadores, o capítulo examina exemplos de aplicativos que usam a rede. Capítulos posteriores explicam como a rede subjacente suporta esses aplicativos.

Um único capítulo não pode cobrir todos os aspectos de programação em rede. Contudo, este demonstrará uma idéia importante:

Um programador pode criar softwares aplicativos da Internet sem entender a rede tecnológica subjacente ou os protocolos de comunicação.

Para demonstrar essa idéia, o capítulo introduz um pequeno conjunto de funções de biblioteca que um programador pode usar e mostra como elas podem ser usadas para escrever aplicativos em rede. Os códigos exemplo para o capítulo estão disponíveis no CD-ROM e na página web, sendo os estudantes encorajados a modificar os exemplos ou escrever outros aplicativos. Capítulos posteriores explicam os detalhes subjacentes e mostram como criar aplicativos sem usar nossas funções de biblioteca.

3.2 Comunicação em rede

Embora uma rede de dados transmita dados de um ponto até outro, a rede, por si só, é passiva. Isso quer dizer que a rede não gera nem comprehende os dados enviados. Na verdade, ela não contém estrutura alguma estrutura para processar informação. Todo o processamento de dados é realizado por programas aplicativos.

Quando aplicativos usam a rede, eles o fazem em pares – o par utiliza a rede meramente para trocar mensagens. Por exemplo, imagine um serviço de banco de dados distribuídos que permita aos usuários acesso remoto a um banco de dados central. Tal serviço requer dois aplicativos, um rodando no computador que tem o banco de dados e o outro rodando num computador remoto. O aplicativo no

computador remoto envia um pedido ao aplicativo no computador com o banco de dados. Quando o pedido chega, o aplicativo rodando no computador com o banco de dados consulta o banco de dados e retorna uma resposta. Apenas os dois aplicativos entendem o formato e o significado da mensagem.

3.3 Arquitetura cliente-servidor

Como um par de programas pode se encontrar numa rede tão larga quanto a Internet? Como a maioria das redes, a Internet utiliza um mecanismo simples: um aplicativo é ligado primeiro e espera que o outro aplicativo faça contato. O segundo aplicativo precisa conhecer a localização na qual o primeiro aplicativo está esperando.

O acordo no qual um aplicativo de rede espera pelo contato de outro é conhecido como *paradigma cliente-servidor* ou como *arquitetura cliente-servidor*. O Capítulo 28 explica a interação cliente-servidor em mais detalhes, e o 29 discute as funções de sockets que os programadores geralmente usam para construir softwares de clientes e servidores. Por enquanto, é suficiente compreender o conceito e a terminologia básica.

O programa que espera pelo contato é chamado de *servidor*, e o que inicia o contato é conhecido como *cliente*. Para iniciar contato, o cliente precisa saber onde o servidor está rodando e especificar a localização para o software de rede.

Como um cliente especifica a localização de um servidor? Na Internet, a localização é dada por um par de identificadores. Conceitualmente, o par consiste em:

(computador, aplicativo)

em que *computador* identifica o computador no qual o servidor está rodando e *aplicativo* identifica um programa aplicativo em particular naquele computador. Softwares de aplicativos representam os dois valores em números binários. Os humanos, porém, nunca precisam lidar diretamente com a representação binária. Em vez disso, os valores também recebem nomes alfabéticos. Humanos digitam os nomes e softwares os traduzem automaticamente para o valor binário correspondente.

3.4 Paradigma da comunicação

A maioria dos aplicativos de Internet segue o mesmo paradigma básico quando se comunica. Dois aplicativos estabelecem comunicação, trocam mensagens para um lado e para o outro, e então terminam a comunicação. Os passos são:

- O aplicativo do servidor é ligado e espera o contato de um cliente.
- O cliente especifica a localização do servidor e requer que uma conexão seja estabelecida.
- Uma vez que a conexão está feita, o cliente e o servidor usam-na para trocar mensagens.
- Depois que param de enviar dados, o cliente e o servidor enviam um *fim de arquivo (end-of-file)* e a conexão é encerrada.

Nossa biblioteca contém funções que tornam cada um desses passos possíveis.

3.5 Um exemplo de API (Application Program Interface)

Até o momento discutimos a interação entre dois aplicativos em um nível conceitual. Vamos considerar agora uma implementação detalhada. Cientistas da Computação utilizam o termo *Application Program Interface (API)* para descrever o conjunto de aplicativos disponíveis a um programador. O API especifica os parâmetros para cada operação, assim como as semânticas.

Para demonstrar a programação de redes, dividimos um API simples para a comunicação de rede. Após descrever o API, vamos considerar outros aplicativos para utilizá-lo. A Figura 3.1 lista as sete funções que possui nosso aplicativo.

Operações	Significados
<code>await_contact</code>	usado por um servidor para esperar pelo contato de um cliente
<code>make_contact</code>	usado por um cliente para contatar um servidor
<code>cname_to_comp</code>	usado para traduzir o nome de um computador para um valor binário interno equivalente
<code>appname_to_appnum</code>	usado para traduzir o nome de um programa para um valor binário interno equivalente
<code>send</code>	usado tanto pelo cliente quanto pelo servidor para enviar dados
<code>recv</code>	usado tanto pelo cliente quanto pelo servidor para receber dados
<code>send_eof</code>	usado tanto pelo cliente quanto pelo servidor depois de terminar de enviar dados

Figura 3.1 Um exemplo de API composto por sete operações. Essas sete funções são suficientes para a maioria dos aplicativos de rede¹.

3.6 Uma olhada intuitiva na API

Um servidor inicia chamando a função `await_contact` para esperar pelo contato de um cliente. O cliente inicia chamando a função `make_contact` para estabelecer contato. Uma vez que o cliente tenha contatado o servidor, eles podem trocar mensagens com `send` e `recv`. Os dois aplicativos devem ser programados para saber se devem enviar ou receber – se ambos tentarem receber sem enviar, ficarão bloqueados para sempre.

Depois que termina de enviar os dados, um aplicativo chama a função `send_eof` para enviar a condição de fim de arquivo (end-of-file). Do outro lado, `recv` retorna um valor zero para indicar que o fim de arquivo foi alcançado. Por exemplo, se o cliente executa `send_eof`, o servidor vai encontrar um valor zero de retorno para sua chamada ao `recv`. Uma vez que ambos os lados tenham invocado `send_eof`, a comunicação é encerrada.

Um exemplo trivial vai ajudar a explicar este exemplo de API. Considere um aplicativo no qual o cliente contata um servidor, envia um único pedido e recebe uma única resposta. A Figura 3.2 ilustra a seqüência de chamadas API que o servidor e o cliente fazem para tal interação.

3.7 Definição de API

Para manter o nosso API independente de algum sistema operacional e software de rede em particular, definimos três tipos de dados e usamos esses tipos através do código. A Figura 3.3 lista os nomes e significados de cada tipo.

Usando os três tipos de nome na Figura 3.3, podemos definir precisamente o API de exemplo. Para cada função, as declarações tipo linguagem C a seguir listam o tipo de cada parâmetro, assim como o tipo dos valores de retorno das funções.

¹ As funções `send` e `recv` são fornecidas diretamente pelo sistema operacional; outras funções no API consistem em rotinas da biblioteca que escrevemos.

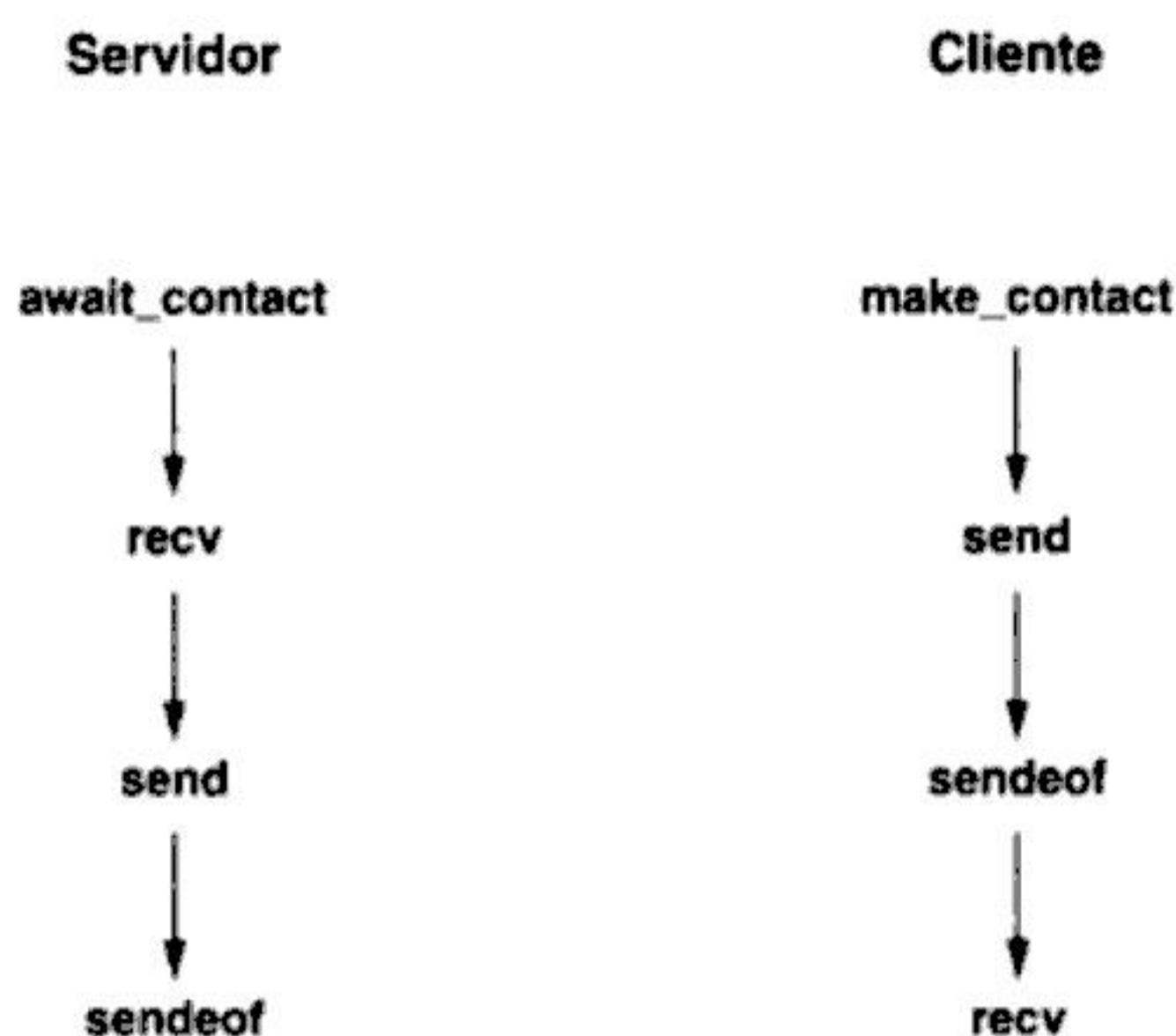


Figura 3.2 Ilustração das chamadas API usadas para uma interação trivial. O cliente envia um pedido e recebe uma resposta.

Nome do tipo	Significado
appnum	Um valor binário para identificar uma aplicação
computer	Um valor binário para identificar um computador
connection	Um valor usado para identificar a conexão entre um cliente e um servidor

Figura 3.3 Os três nomes de tipo usados em nosso exemplo API. Em um dado computador, esses tipos são definidos para ser inteiros e de tamanho específico.

3.7.1 A função `await_contact`

Um servidor chama a função `await_contact` para esperar o contato de um cliente.

| connection await_contact(appnum a)

A chamada pega um valor do tipo `appnum` e retorna um valor do tipo `connection`. O parâmetro especifica um número que identifica o aplicativo do servidor; o cliente precisa especificar o mesmo número quando estiver contatando o servidor. O servidor usa o valor de retorno (tipo `connection`) para transferir dados.

3.7.2 A função `make_contact`

Um cliente chama a função `make_contact` para estabelecer contato com o servidor.

| connection make_contact(computer c, appnum a)

Uma chamada necessita dois parâmetros para identificar um computador no qual o servidor esteja rodando e o número do aplicativo que o servidor está usando naquele computador. O cliente utiliza o valor de retorno, que é do tipo `connection`, para transferir dados.

3.7.3 A função `appname_to_appnum`

Tanto clientes quanto servidores utilizam `appname_to_appnum` para traduzir o nome de um serviço legível por humanos para um valor binário interno. Os nomes de serviço são padronizados através da Internet (p. ex.: www denota *World Wide Web*).

`| appnum appname_to_appnum(char *a)`

A chamada utiliza um parâmetro do tipo *string* (utiliza a declaração `char *` para denotar um *string*) e retorna um valor binário equivalente do tipo `appnum`.

3.7.4 A função `cname_to_comp`

Os clientes chamam `cname_to_comp` para converter um nome de computador legível por humanos para um valor binário interno.

`| computer cname_to_comp(char *c)`

A chamada toma um parâmetro do tipo *string* (`char *`) e retorna um valor binário equivalente do tipo `computer`.

3.7.5 A função `send`

Tanto os clientes quanto os servidores usam `send` para transferir dados através da rede.

`| int send(connection con, char *buffer, int lenght, int flags)`

A chamada toma quatro parâmetros. O primeiro identifica uma conexão previamente estabelecida com `await_contact` ou `make_contact`; o segundo é o endereço de um buffer contendo dados a serem enviados; o terceiro dá o tamanho dos dados em bytes (oito bits); e o quarto é zero para uma transferência normal. `Send` retorna o número de bytes transferido, ou um valor negativo, caso um erro tenha ocorrido. Veja também `send_eof`, usado para enviar um *fim de arquivo* depois que todos os dados foram enviados.

3.7.6 As funções `recv` e `recvln`

Tanto clientes quanto servidores usam `recv` para acessar os dados que chegam através da rede.

`| int recv(connection con, char *buffer, int lenght, int flags)`

A chamada toma quatro parâmetros. O primeiro identifica uma conexão previamente estabelecida com `await_contact` ou `make_contact`; o segundo é o endereço de um buffer para onde os dados devem ser enviados; o terceiro dá o tamanho do buffer em bytes (oito bits); e o quarto é zero para transferências normais. `Recv` retorna o número de bytes que foram colocados no buffer, zero para indicar que o *fim de arquivo* foi alcançado, ou um valor negativo, para indicar que um erro ocorreu.

No código de exemplo, também usamos uma função de biblioteca `recvln` que repetidamente chama `recv` até que uma linha inteira do texto tenha sido recebida. A definição de `recvln` é:

`| int recvln(connection con, char *buffer, int lenght)`



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

```

    /* wait for a connection from an echo client */

    conn = await_contact((appnum) atoi(argv[1]));
    if (conn < 0)
        exit(1);

    /* iterate, echoing all data received until end of file */

    while((len = recv(conn, buff, BUFFSIZE, 0)) > 0)
        (void) send(conn, buff, len, 0);
    send_eof(conn);
    return 0;
}

```

Como foi visto, o servidor toma uma única linha de comando como parâmetro para especificar o número da aplicação a ser utilizada. Em linguagem C, linhas de comando como parâmetros são passadas para o programa como um vetor de strings (*argv*), junto com um contador inteiro de parâmetros (*argc*). O código extrai a linha de comando como parâmetro do *argv[1]* e chama a função padrão da linguagem C *atoi* para converter o valor de um string ASCII para binário. Ele então passa o resultado como um parâmetro para *await_contact*. Uma vez que a chamada para *await_contact* retorna um valor, o servidor chama repetidamente *recv* para receber dados do cliente e *send* para transmitir os mesmos dados de volta. A interação termina quando *recv* encontra uma resposta de fim de arquivo e retorna um valor zero. Nessa hora, o servidor envia um aviso de fim de arquivo e desconecta.

3.8.2 Exemplo de código para um cliente de eco

O arquivo *echoclient.c* contém o código para um aplicativo de cliente de eco.

```

/* echoclient.c */

#include <stdlib.h>
#include <stdio.h>
#include <cnaiaapi.h>

#define BUFFSIZE          256
#define INPUT_PROMPT      "Input  > "
#define RECEIVED_PROMPT   "Received> "

int readln(char *, int);

/*
 * Program: echoclient
 * Purpose: contact echoserver, send user input and print server response
 * Usage:   echoclient <compname> [appnum]
 * Note:    Appnum is optional. If not specified the standard echo appnum
 *          (7) is used.
 *
 */
int
main(int argc, char *argv[])
{

```

```
computer      comp;
appnum        app;
connection    conn;
char          buff[BUFFSIZE];
int           expect, received, len;

if (argc < 2 || argc > 3) {
    (void) fprintf(stderr, "usage: %s <compname> [appnum]\n",
                   argv[0]);
    exit(1);
}

/* convert the arguments to binary format comp and appnum */

comp = cname_to_comp(argv[1]);
if (comp == -1)
    exit(1);

if (argc == 3)
    app = (appnum) atoi(argv[2]);
else
    if ((app = appname_to_appnum("echo")) == -1)
        exit(1);

/* form a connection with the echoserver */

conn = make_contact(comp, app);
if (conn < 0)
    exit(1);

(void) printf(INPUT_PROMPT);
(void) fflush(stdout);

/* iterate: read input from the user, send to the server,      */
/*          receive reply from the server, and display for user */
/*          while((len = readln(buff, BUFFSIZE)) > 0) { */

    /* send the input to the echoserver */

    (void) send(conn, buff, len, 0);
    (void) printf(RECEIVED_PROMPT);
    (void) fflush(stdout);

    /* read and print same no. of bytes from echo server */

    expect = len;
    for (received = 0; received < expect;) {
        len = recv(conn, buff, (expect - received) < BUFFSIZE ?
                    (expect - received): BUFFSIZE, 0);
        if (len < 0) {
            send_eof(conn);
            return 1;
        }
        (void) write(STDOUT_FILENO, buff, len);
        received += len;
    }
}
```

```
    }

    (void) printf("\n");
    (void) printf(INPUT_PROMPT);
    (void) fflush(stdout);
}

/* iteration ends when EOF found on stdin */

(void) send_eof(conn);
(void) printf("\n");
return 0;
}
```

O programa de cliente toma um ou dois parâmetros. O primeiro parâmetro especifica o nome do computador no qual o servidor está rodando. Se presente, o segundo parâmetro especifica o número do aplicativo que o servidor está usando. Se o segundo parâmetro não está presente, o cliente chama *appname_to_appnum* com o parâmetro *echo*.

Após converter os parâmetros para a forma binária, o cliente os passa para a função *make_contact*, que contata o servidor. Uma vez que o contato tenha sido estabelecido, o cliente espera pelo usuário e entra em um laço que lê a linha de entrada, envia a linha para o servidor, lê a resposta do servidor e imprime-a para o usuário seguida de uma nova espera pelo usuário. Quando o cliente alcança o fim da entrada (p. ex.: *readln* retorna ao valor zero), ele chama *send_eof* para informar ao servidor e então desconecta.

Diversos detalhes complicam o código. Primeiro, o cliente chama a função *readln* para ler uma linha de entrada. Segundo, o cliente testa o valor de retorno para cada chamada de função e desconecta quando um valor indica que um erro ocorreu. Terceiro, o cliente chama *fflush* para garantir que a saída é mostrada imediatamente, em vez de ser armazenada num buffer. Quarto, e mais significativo, o cliente não publica uma simples chamada para *recv* cada vez que recebe uma chamada do servidor. Em vez disso, ele entra num laço que chama *recv* repetidamente até que tenha recebido tantos bytes quanto foram enviados.

O uso de múltiplas chamadas para *recv* levanta um ponto crucial sobre nosso API:

Um receptor não pode presumir que os dados vão chegar em pedaços do mesmo tamanho daqueles enviados; uma chamada para recv pode retornar menos dados do que foram enviados em uma chamada para send.

Capítulos posteriores explicarão por que *recv* se comporta dessa forma: redes dividem os dados em pequenos "pacotes". Por isso, um aplicativo pode receber os dados de um pacote de cada vez. Surpreendentemente, o oposto também é verdadeiro. Mesmo se quem envia chama *send* repetidamente, o software de rede pode receber dados de pacotes antes de o aplicativo chamar *recv*. Em tais casos, *recv* irá retornar todos os dados de uma só vez.

3.9 Código para um aplicativo de chat

O segundo aplicativo considerado é uma forma simplificada de um ambiente de chat. Na Internet, o chat permite a comunicação de um grupo de usuários pela entrada de mensagens de texto que são visualizadas na tela de cada um. Nossa software fornece uma versão simplificada de chats que trabalham entre um único par de usuários – quando um usuário entra o texto, este é mostrado na tela do outro usuário, e vice-versa. Além disso, como o aplicativo de eco descrito anteriormente, nosso software de chat pode ser utilizado entre quaisquer computadores conectados à Internet. Um usuário inicia escolhendo o número do aplicativo e rodando o servidor. Suponha que um usuário no computador *excalibur.cs.purdue.edu* execute o servidor:

```
chatserver 25000
```

Um usuário em outro computador pode invocar o cliente, que contata o servidor:

```
chatclient excalibur.cs.purdue.Edu 25000
```

Para manter o código tão curto quanto possível, escolhemos um esquema que exige dos usuários entrar o texto em turnos. Quando se espera que o usuário entre com uma linha de texto, servidor e cliente emitem um prompt (aviso de espera). O usuário no lado do cliente recebe o aviso pela primeira entrada. Quando uma linha de texto é recebida, o cliente a envia para o servidor e as funções se invertem. Os usuários se alternam enviando textos até que um deles envie uma mensagem de fim de arquivo.

O código é simples. O servidor inicia esperando pelo contato do cliente. Ele então entra num laço no qual obtém e mostra uma linha de texto do cliente, dá um aviso de espera para o usuário local, lê uma linha de entrada a partir do teclado e envia a linha para o lado do cliente. Então, até receber um aviso de fim de arquivo, o servidor itera entre mostrar a saída do cliente e enviar uma entrada do teclado para o cliente.

O cliente inicia por contatar o servidor. Uma vez que a comunicação é estabelecida, o cliente também entra em laço. Durante cada iteração, ele manda um aviso de espera ao usuário local para entrar uma linha de texto, lê a linha a partir do teclado, envia a linha para o servidor e então recebe e mostra a linha de texto do servidor. Então, o cliente continua a alternar entre enviar uma linha de texto que o usuário entrou e mostrar uma linha de texto do servidor.

3.9.1 Código de exemplo para um servidor de chat

O arquivo *chatserver.c* contém o código para o servidor de *chat*.

```
/* chatserver.c */

#include <stdlib.h>
#include <stdio.h>
#include <cnaiaapi.h>

#define BUFFSIZE          256
#define INPUT_PROMPT      "Input    > "
#define RECEIVED_PROMPT   "Received> "

int recvln(connection, char *, int);
int readln(char *, int);

/*
 * Program: chatserver
 * Purpose: wait for a connection from a chatclient & allow users to chat
 * Usage:   chatserver <appnum>
 *
 */
int
main(int argc, char *argv[])
{
    connection          conn;
    int                 len;
    char                buff[BUFFSIZE];
```

```

if (argc != 2) {
    (void) fprintf(stderr, "usage: %s <appnum>\n", argv[0]);
    exit(1);
}

(void) printf("Chat Server Waiting For Connection.\n");

/* wait for a connection from a chatclient */

conn = await_contact((appnum) atoi(argv[1]));
if (conn < 0)
    exit(1);

(void) printf("Chat Connection Established.\n");
/* iterate, reading from the client and the local user */

while((len = recvln(conn, buff, BUFFSIZE)) > 0) {
    (void) printf(RECEIVED_PROMPT);
    (void) fflush(stdout);
    (void) write(STDOUT_FILENO, buff, len);

    /* send a line to the chatclient */

    (void) printf(INPUT_PROMPT);
    (void) fflush(stdout);
    if ((len = readln(buff, BUFFSIZE)) < 1)
        break;
    buff[len - 1] = '\n';
    (void) send(conn, buff, len, 0);
}

/* iteration ends when EOF found on stdin or chat connection */

(void) send_eof(conn);
(void) printf("\nChat Connection Closed.\n\n");
return 0;
}

```

As funções *recvln* e *readln* simplificam o código – cada uma consiste em um laço que itera até uma linha inteira ou um fim de arquivo a ser encontrado. *Recvln* chama *recv* para receber dados de uma conexão de rede e *readln* chama *read* para ler os caracteres de um teclado.

A estrutura do servidor de chat como um todo é similar ao servidor de eco, pois espera uma única linha de comando como parâmetro, que é o número do aplicativo a ser utilizado. Uma vez que chega o contato do cliente, o servidor de chat imprime a mensagem para o usuário local e entra em um laço. A cada iteração, o servidor recebe a linha de texto através da conexão de rede, imprime a linha na tela do usuário, lê a linha de entrada no teclado e envia a linha através da rede. Quando detecta um fim de arquivo, o servidor envia outro fim de arquivo e desconecta.

3.9.2 Código de exemplo para um cliente de chat

O arquivo *chatclient.c* contém o código para o cliente de chat.

```
/* chatclient.c */
```



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

```
/* contact the web server */

conn = make_contact(comp, app);
if (conn < 0)
    exit(1);

/* send an HTTP/1.0 request to the webserver */

len = sprintf(buff, "GET %s HTTP/1.0\r\n\r\n", argv[2]);
(void) send(conn, buff, len, 0);

/* dump all data received from the server to stdout */

while((len = recv(conn, buff, BUFFSIZE, 0)) > 0)
    (void) write(STDOUT_FILENO, buff, len);

return 0;
}
```

O código do cliente é extremamente simples – após estabelecer a comunicação com o servidor Web, ele envia um pedido, que precisa ter o seguinte formato servidor⁴

GET /path HTTP/1.0 CRLF CRLF

onde *path* denota o nome de um item como *index.html* e CRLF denota os dois caracteres *carriage return* e *line feed*. Depois de enviar o pedido, o cliente recebe e imprime a saída do servidor.

3.10.2 Código de exemplo para um servidor de Web

O arquivo *webserver.c* contém o código para um servidor Web (em miniatura):

```
/* webserver.c */

#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <cnaiaapi.h>

#if defined(LINUX) || defined(SOLARIS)
#include <sys/time.h>
#endif

#define BUFFSIZE      256
#define SERVER_NAME   "CNAI Demo Web Server"

#define ERROR_400      "<head></head><body><html><h1>Error 400</h1><p>The server couldn't understand your request.</html></body>\n"

#define ERROR_404      "<head></head><body><html><h1>Error 404</h1><p>Document not found.</html></body>\n"

#define HOME_PAGE      "<head></head><body><html><h1>Welcome to the CNAI Demo Server</h1><p>Why not visit: <ul><li><a href=\"http://netbook.cs.pu\
```

⁴ Capítulos posteriores explicam o formato em mais detalhes.

```
rdue.edu\">Netbook Home Page</a><li><a href=\"http://www.comerbooks.com\">  
>Comer Books Home Page<a></ul></html></body>\n"  
  
#define TIME_PAGE      "<head></head><body><html><h1>The current date is\\n<pre>"  
: %s</h1></pre></body></html>\n"  
  
int      recvln(connection, char *, int);  
void    send_head(connection, int, int);  
  
/*-----  
 *  
 * Program: webserver  
 * Purpose: serve hard-coded webpages to web clients  
 * Usage:   webserver <appnum>  
 *  
 *-----  
 */  
int  
main(int argc, char *argv[])  
{  
  
    connection conn;  
    int n;  
    char buff[BUFFSIZE], cmd[16], path[64], vers[16];  
    char *timestr;  
#if defined(LINUX) || defined(SOLARIS)  
    struct timeval tv;  
#elif defined(WIN32)  
    time_t tv;  
#endif  
  
    if (argc != 2) {  
        (void) fprintf(stderr, "usage: %s <appnum>\\n", argv[0]);  
        exit(1);  
    }  
  
    while(1) {  
  
        /* wait for contact from a client on specified appnum */  
  
        conn = await_contact((appnum) atoi(argv[1]));  
        if (conn < 0)  
            exit(1);  
  
        /* read and parse the request line */  
  
        n = recvln(conn, buff, BUFFSIZE);  
        sscanf(buff, "%s %s %s", cmd, path, vers);  
  
        /* skip all headers - read until we get \\r\\n alone */  
  
        while((n = recvln(conn, buff, BUFFSIZE)) > 0) {  
            if (n == 2 && buff[0] == '\\r' && buff[1] == '\\n')  
                break;  
        }  
    }  
}
```

```
    /* check for unexpected end of file */

    if (n < 1) {
        (void) send_eof(conn);
        continue;
    }

    /* check for a request that we cannot understand */

    if (strcmp(cmd, "GET") || (strcmp(vers, "HTTP/1.0") &&
                               strcmp(vers, "HTTP/1.1"))) {
        send_head(conn, 400, strlen(ERROR_400));
        (void) send(conn, ERROR_400, strlen(ERROR_400), 0);
        (void) send_eof(conn);
        continue;
    }

    /* send the requested web page or a "not found" error */

    if (strcmp(path, "/") == 0) {
        send_head(conn, 200, strlen(HOME_PAGE));
        (void) send(conn, HOME_PAGE, strlen(HOME_PAGE), 0);
    } else if (strcmp(path, "/time") == 0) {
#ifndef defined(LINUX) || defined(SOLARIS)
        gettimeofday(&tv, NULL);
        timestr = ctime(&tv.tv_sec);
#endif defined(WIN32)
        time(&tv);
        timestr = ctime(&tv);
#endif
        (void) sprintf(buff, TIME_PAGE, timestr);
        send_head(conn, 200, strlen(buff));
        (void) send(conn, buff, strlen(buff), 0);
    } else { /* not found */
        send_head(conn, 404, strlen(ERROR_404));
        (void) send(conn, ERROR_404, strlen(ERROR_404), 0);
    }
    (void) send_eof(conn);
}

/*
 * send_head - send an HTTP 1.0 header with given status and content-len
 */
void
send_head(connection conn, int stat, int len)
{
    char *statstr, buff[BUFFSIZE];

    /* convert the status code to a string */

    switch(stat) {
    case 200:
        statstr = "OK";
        break;
```

```

        case 400:
            statstr = "Bad Request";
            break;
        case 404:
            statstr = "Not Found";
            break;
        default:
            statstr = "Unknown";
            break;
    }

/*
 * send an HTTP/1.0 response with Server, Content-Length,
 * and Content-Type headers.
 */

(void) sprintf(buff, "HTTP/1.0 %d %s\r\n", stat, statstr);
(void) send(conn, buff, strlen(buff), 0);

(void) sprintf(buff, "Server: %s\r\n", SERVER_NAME);
(void) send(conn, buff, strlen(buff), 0);

(void) sprintf(buff, "Content-Length: %d\r\n", len);
(void) send(conn, buff, strlen(buff), 0);

(void) sprintf(buff, "Content-Type: text/html\r\n");
(void) send(conn, buff, strlen(buff), 0);

(void) sprintf(buff, "\r\n");
(void) send(conn, buff, strlen(buff), 0);
}

```

Apesar de o servidor da Web parecer mais complexo que os exemplos anteriores, a maior parte da complexidade resulta dos detalhes Web, em vez dos detalhes de rede. Além de ler e interpretar (parsing) um pedido, o servidor precisa enviar um “header” (cabeçalho) e os dados na resposta. O cabeçalho consiste em algumas linhas de texto encerradas pelos caracteres carriage return e line feed. As linhas do cabeçalho são no formato:

```

HTTP/1.0 status status_string CRLF
Server: CNAI Demo Server CRLF
Content-Length: datasize CRLF
Content-Type: text/html CRLF
CRLF

```

onde *datasize* significa o tamanho dos dados que seguem, medidos em bytes.

O procedimento *send_head* lida com a tarefa de gerar um cabeçalho. Quando *send_head* é chamado, o parâmetro *stat* contém um código do estado em inteiro (integer) e o parâmetro *len* especifica o tamanho do conteúdo. A declaração *switch* usa o código para escolher uma mensagem de texto apropriada, que é atribuída para a variável *statstr*. *Send_head* usa a função de linguagem C *sprintf* para gerar o cabeçalho completo num buffer, e então chama *send* para transmitir as linhas do cabeçalho através da conexão para o cliente.

O código também se complica por lidar com erros – mensagens de erro precisam ser enviadas de uma forma que o navegador comprehenda. Se um pedido for formulado incorretamente, o servi-

dor gera uma mensagem de erro 400; Se o item especificado no pedido não puder ser encontrado (p.ex., o *path* está incorreto), o servidor gera uma mensagem 404.

Este servidor de Web difere dos exemplos anteriores de uma forma significativa: o programa do servidor não desconecta depois de satisfazer um pedido. Em vez disso, o servidor continua rodando, pronto para aceitar outros pedidos. Isso quer dizer que o programa do servidor consiste em um laço infinito que chama *await_contact* para esperar pelo contato de um cliente. Quando o contato chega, o servidor chama *recvln* para receber um pedido e chama *send* para enviar uma resposta. O servidor então volta para o início do laço para esperar pelo próximo contato. Portanto, uma vez que tenha sido iniciado, o servidor roda para sempre, assim como servidor de Web comercial.

3.11 Gerenciando múltiplas conexões com a função select

Embora o exemplo de API suporte a interação de um para um entre cliente e servidor, o API não suporta a interação de um para vários. Para entender por que, considere múltiplas conexões. Para criar tais conexões, um único programa de aplicativo precisa chamar *make_contact* várias vezes, especificando um *computador* e *appnum* para cada chamada. Uma vez que as conexões tenham sido estabelecidas, o aplicativo não pode saber qual deles vai receber a mensagem primeiro. O aplicativo não pode usar *recv* porque a chamada estará bloqueada até todos os dados chegarem. Muitos sistemas operacionais incluem uma função chamada *select* que resolve o problema de lidar com múltiplas conexões. Conceitualmente, a função *select* checa um conjunto de conexões. A chamada é bloqueada até que pelo menos uma das conexões especificadas tenha recebido os dados. A chamada então retorna um valor que avisa quais das conexões receberam os dados (e.g.: conexões para as quais a função *recv* não será bloqueada).

Como um exemplo, considere um aplicativo que precise receber pedidos e enviar respostas para duas conexões. Tal aplicativo pode ter a seguinte forma geral:

```

Call make_contact to form connection 1;
Call make_contact to form connection 2;
Repeat forever {
    Call select to determine which connection is ready
    If (connection 1 is ready) {
        Call recv to read request from connection 1;
        Compute response to request;
        Call send to send response over connection 1;
    } if (connection 2 is ready) {
        Call recv to read request from connection 2;
        Compute response to request;
        Call send to send response over connection 2;
    }
}

```

3.12 Resumo

É possível para um programador criar aplicativos de rede que operem através da Internet global sem entender como as redes funcionam ou como as tecnologias subjacentes carregam os dados entre computadores. O programador precisa receber um conjunto de funções de alto nível para formar um API (*Application Program Interface*). Este capítulo apresenta uma rede API que contém apenas funções primitivas e analisa exemplos de aplicativos que mostram que este API é suficiente para construir softwares que interajam corretamente com softwares comerciais.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

Meios de Transmissão

Os fundamentos dos meios de transmissão, dos sinais, dos bits, das portadoras e dos modems

Sumário do Capítulo

- 4.1 Introdução 71**
- 4.2 Fios de Cobre 71**
- 4.3 Fibras de Vidro 73**
- 4.4 Rádio 73**
- 4.5 Satélites 74**
- 4.6 Satélites Geossíncronos 75**
- 4.7 Satélites de Órbita Baixa da Terra 75**
- 4.8 Arrays de Satélites de Órbita Baixa da Terra 75**
- 4.9 Microonda 76**
- 4.10 Infravermelho 76**
- 4.11 Luz de Laser 76**
- 4.12 Resumo 77**

Meios de Transmissão

4.1 Introdução

No nível mais baixo, toda a comunicação entre computadores envolve codificar dados em uma forma de energia e enviar essa energia através de um meio de transmissão. Por exemplo, a corrente elétrica pode ser usada para transferir dados através de um fio, ou as ondas de rádio podem ser usadas para carregar dados através do ar. Uma vez que os dispositivos de hardware conectados a um computador executam a codificação e a decodificação dos dados, os programadores e os usuários não precisam conhecer os detalhes da transmissão. Entretanto, já que um papel principal do software de comunicação é tratar os erros e as falhas que surgem no hardware subjacente, compreender tal software requer o conhecimento de alguns conceitos básicos sobre transmissão de dados.

Esta seção cobre os princípios da transmissão de dados. O primeiro capítulo examina os meios usados para a transmissão em sistemas de rede modernos. Os dois capítulos seguintes explicam como os dados podem ser transferidos através de tais meios. As seções posteriores explicam como a transmissão forma a base da interligação de dados em rede.

4.2 Fios de cobre

As redes de computadores convencionais usam fios como o meio primário para conectar computadores porque são baratos e fáceis de instalar. As redes de computadores usam quase exclusivamente fios de cobre porque sua baixa resistência à corrente elétrica faz com que os sinais possam viajar mais longe. Assim, os profissionais de rede usam, às vezes, o termo *cobre* como sinônimo para *fio*.

O tipo de fiação usado em redes de computadores é escolhido para minimizar a interferência. A interferência surge porque um sinal elétrico que viaja através de um fio age como uma estação de rádio em miniatura – o fio emite um pouco de energia eletromagnética, que pode viajar através do ar. Além disso, sempre que encontra outro fio, uma onda eletromagnética gera uma corrente elétrica pequena no fio. A quantidade de corrente gerada depende da força da onda eletromagnética e da posição física do fio. Geralmente, os fios não chegam perto o suficiente para fazer da interferência um problema. Se dois fios estão dispostos um do outro em um ângulo reto e um sinal

passa através de um deles, a corrente gerada no outro é quase indetectável. Entretanto, quando ambos são colocados juntos e em paralelo, um sinal forte enviado em um gerará um sinal similar no outro. Já que os computadores não podem distinguir entre sinais gerados acidentalmente e as transmissões normais, a corrente gerada pode ser forte o bastante para transtornar ou impedir uma comunicação normal. Infelizmente, o problema da interferência é sério porque os fios que compreendem uma rede de dados são freqüentemente colocados em paralelo com muitos outros. Por exemplo, os fios de um computador podem encontrar-se ao lado dos fios de outros computadores ou dos fios para outras redes.

Para minimizar a interferência, as redes usam um de três tipos básicos de fiação:

- par trançado descoberto (UTP)
- cabo coaxial
- par trançado coberto (STP)

A fiação trançada do par é usada também por sistemas de telefonia. O termo se deve ao fato de que cada fio é revestido com um material isolante (por exemplo, plástico), e então um par dos fios é torcido junto, tal como apresentado na Figura 4.1.

As torções simples mudam as propriedades elétricas do fio e ajudam o mesmo a se tornar apropriado para o uso em uma rede. Em primeiro lugar, uma vez que limitam a energia eletromagnética que o fio emite, as torções ajudam a impedir que as correntes elétricas no fio irradiem energia que interfere em outros fios. Em segundo, já que fazem o par dos fios menos suscetível à energia eletromagnética, as torções ajudam a impedir que os sinais em outros fios interfiram no par.

O segundo tipo de fiação de cobre usado nas redes é o cabo coaxial (coax), o mesmo usado para a televisão a cabo. O coaxial fornece proteção ainda maior contra interferência do que o par trançado. Em vez de trançar fios um ao redor do outro para limitar a interferência, um cabo coaxial consiste em um único fio cercado por um protetor de metal mais pesado, como ilustrado na Figura 4.2.

O protetor de metal pesado em um cabo coaxial forma um cilindro flexível, em torno do fio interno, que fornece uma barreira à radiação eletromagnética. A barreira isola o fio interno de duas maneiras: protege-o da energia eletromagnética entrante que poderia causar a interferência e evita que sinais no fio interno irradiem energia eletromagnética que poderia afetar outros fios. Uma vez



Figura 4.1 Ilustração da fiação com par trançado. Uma cobertura plástica na superfície de cada fio evita que o metal de um fio toque o metal do outro. As torções ajudam a reduzir a interferência.

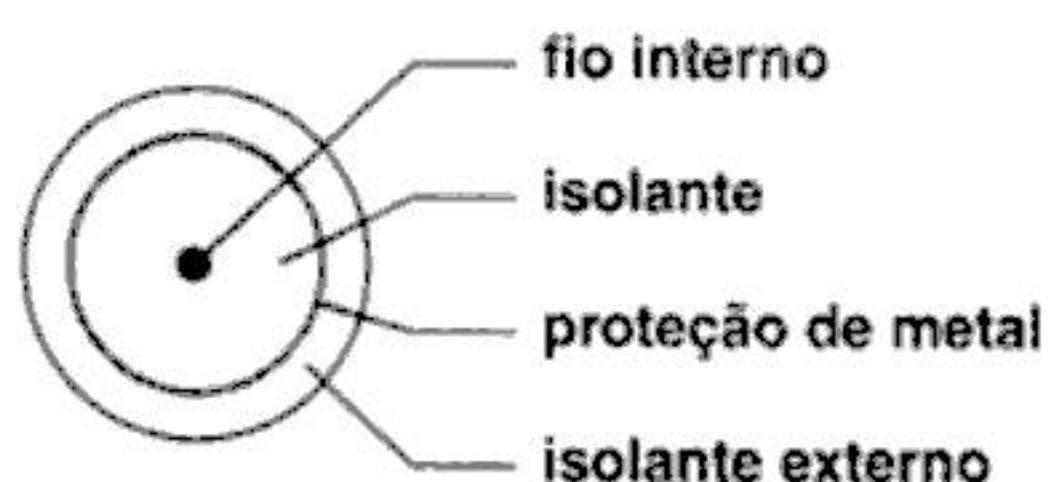


Figura 4.2 Corte transversal aumentado de um cabo coaxial com as partes mais importantes identificadas. Embora um cabo coaxial seja mais duro que um único fio, ele pode ser dobrado.

que cerca o centro do fio uniformemente em todos os lados, o protetor em um cabo coaxial é especialmente eficaz. O cabo pode ser colocado em paralelo com outros ou dobrado e torcido em torno dos cantos. O protetor permanece sempre no lugar.

A idéia de usar um protetor para os fios foi aplicada também ao par trançado. Um cabo *par trançado protegido* consiste em um par de fios cercado por um protetor de metal. Cada fio é revestido com um material isolante, de forma que o metal em um fio não toque o metal em outro – o protetor forma uma barreira que impede a radiação eletromagnética de entrar ou escapar. A proteção adicional fornecida pelo cabeamento com cabo coaxial ou par trançado protegido é freqüentemente usada quando os fios de uma rede passam perto de equipamento que gera campos elétricos ou magnéticos fortes (por exemplo, um grande condicionador de ar).

4.3 Fibras de vidro

As redes de computadores usam também fibras de vidro flexíveis para transmitir dados. Conhecido como *fibra óptica*, o meio usa a luz para transportar dados. A fibra de vidro em miniatura é revestida de plástico, que permite que a fibra se dobre sem quebrar¹. Um transmissor em uma extremidade de uma fibra usa um *diodo emissor de luz* (*light emitting diode, LED*) ou um *laser* para enviar pulsos de luz pela fibra. Um receptor no extremo oposto usa um transistor sensível à luz para detectar os pulsos.

As fibras ópticas têm quatro vantagens principais sobre os fios. Em primeiro lugar, como usam luz, elas não causam interferência elétrica em outros cabos, nem são suscetíveis à interferência elétrica. Em segundo lugar, como podem ser fabricadas para refletir a maioria da luz interna, podem carregar um pulso de luz muito mais longe do que um fio de cobre pode carregar um sinal. Em terceiro lugar, já que a luz pode codificar mais informação do que sinais elétricos, fibras ópticas podem carregar mais informação do que um fio. Em quarto lugar, ao contrário da eletricidade, que requer sempre um par de fios conectado em um circuito completo, a luz pode viajar de um computador a outro sobre uma única fibra.

Apesar de suas vantagens, as fibras ópticas têm algumas desvantagens. Primeiro, a instalação de uma fibra requer um equipamento especial que faça o polimento das extremidades para permitir que a luz passe completamente. Segundo, se uma fibra quebrar dentro do revestimento plástico (por exemplo, sendo dobrada em um ângulo reto), encontrar onde ocorreu o problema é difícil. Terceiro, reparar uma fibra quebrada é complicado porque equipamento especial é necessário para juntar duas fibras de modo que a luz possa passar através da junção.

4.4 Rádio

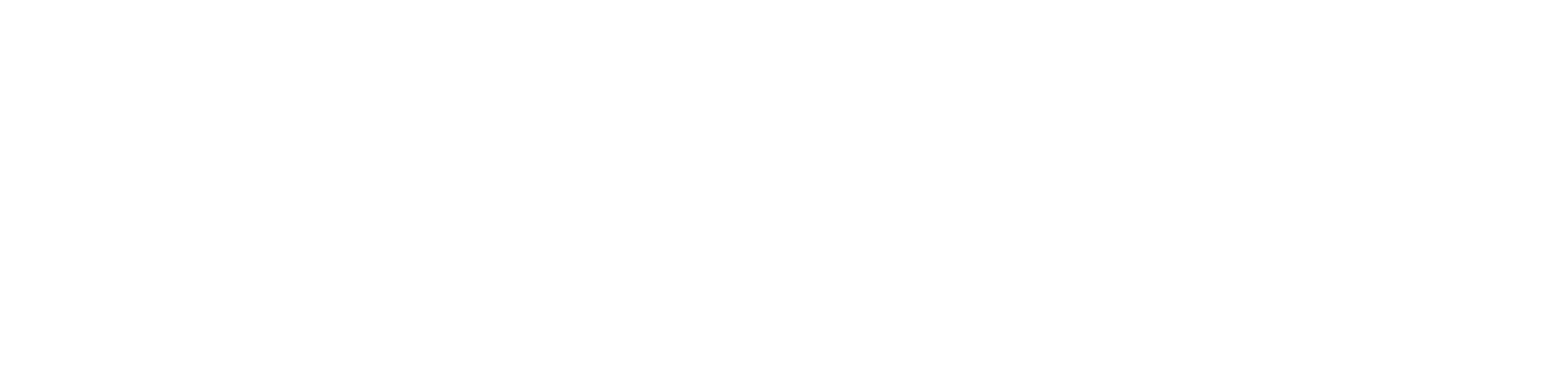
Além de seus usos para a transmissão pública dos programas de rádio e de televisão e para uma comunicação privada com dispositivos, como telefones portáteis, a radiação eletromagnética pode ser usada para transmitir dados de computador. Informalmente, diz-se que uma rede que usa ondas de rádio eletromagnéticas opera na *frequência de rádio*, e as transmissões são chamadas de *transmissões RF*. Ao contrário das redes que usam fios ou fibras ópticas, as que usam transmissões de transmissões RF não requerem uma conexão física direta entre computadores. Em vez disso, cada computador participante está anexo a uma antena, que pode tanto transmitir como receber RF.

Fisicamente, as antenas usadas com redes RF podem ser grandes ou pequenas, dependendo do alcance desejado. Uma antena projetada para propagar sinais por diversos quilômetros através da

¹ Embora uma fibra óptica não possa ser dobrada em um ângulo reto, ela pode formar um círculo com raio menor do que duas polegadas.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

Além dos transponders usados para a comunicação com as estações da Terra, um array de satélites de órbita baixa contém equipamento de rádio usado para comunicar-se com outros satélites no array. Enquanto se movem através de suas órbitas, os satélites se comunicam uns com os outros e concordam no encaminhamento de dados. Por exemplo, suponha que, em um dado momento, um satélite viajando sobre a Europa receba uma transmissão de uma estação terrestre na Alemanha, destinada para um local dos Estados Unidos. O satélite receptor poderia repassar uma transmissão a outro satélite, que a repassaria a um terceiro, que poderia alcançar uma estação terrestre nos Estados Unidos próxima ao destino. Com o passar do tempo, os satélites vão adiante e os novos satélites tomam seu lugar, significando que mais tarde uma transmissão da Alemanha para os Estados Unidos poderia ter que atravessar três outros satélites no array.

4.9 Microonda

A radiação eletromagnética além da faixa de freqüência usada por rádio e televisão pode ser usada também para transportar informações. Em particular, muitas empresas de telefonia interurbana usam transmissões de microonda para carregar conversas telefônicas. Algumas empresas de grande porte instalaram também sistemas de comunicação de microonda como parte do sistema de rede da empresa.

Embora as microondas sejam meramente uma versão das ondas de rádio com freqüência mais elevada, elas se comportam de maneira diferente. Em vez de transmitir em todas as direções, uma transmissão de microondas pode ser apontada em uma única direção, impedindo que outros interceptem o sinal. Além disso, a transmissão de microondas pode carregar mais informações do que transmissões RF de freqüência mais baixa. Entretanto, já que as microondas não podem penetrar estruturas de metal, a transmissão de microondas trabalha melhor quando há um trajeto desobstruído entre o transmissor e o receptor. Como consequência, a maioria das instalações baseadas em microondas consistem em duas torres, mais altas do que os edifícios e a vegetação circundantes, cada uma com um transmissor apontado diretamente para um receptor de microondas.

4.10 Infravermelho

Os controles remotos sem fio, usados com dispositivos como aparelhos de televisão e de som, comunicam-se através de transmissões em *infravermelho*. O infravermelho é limitado a uma área pequena (por exemplo, uma única sala) e geralmente exige que o transmissor esteja apontado para o receptor. O hardware infravermelho é barato, se comparado a outros mecanismos, e não requer antena.

As redes de computadores podem usar a tecnologia infravermelha para a transmissão de dados. Por exemplo, é possível equipar uma sala grande com uma única conexão infravermelha que forneça acesso à rede a todos os computadores na sala. Os computadores podem permanecer em contato com a rede enquanto são movidos dentro da sala. As redes de infravermelho são especialmente convenientes para computadores pequenos e portáteis, porque o infravermelho oferece as vantagens de comunicação sem fio sem necessitar do uso de antenas. Assim, um computador portátil que use infravermelho pode ter todo o hardware de comunicação embutido.

4.11 Luz de laser

Já foi mencionado que a luz pode ser usada para comunicação através de fibras ópticas. Um feixe de luz pode também ser usado para carregar dados através do ar. Como um sistema de comunicação de microondas, uma comunicação que use luz consiste em dois locais, um que possui um transmissor e outro que possui um receptor. O equipamento de comunicação é montado em uma posição fixa, freqüentemente em uma torre, e alinhado de forma que o transmissor em uma posição envie seu feixe de luz diretamente ao receptor na outra. O transmissor usa um *laser* para gerar o feixe de luz, pois um feixe de laser coerente permanecerá focalizado por uma longa distância.

Como uma transmissão de microondas, a luz de um laser deve viajar em uma linha reta e não deve ser obstruída. Infelizmente, um feixe de laser não pode penetrar na vegetação ou em condições climáticas como neve e névoa. Assim, a transmissão via laser tem uso limitado.

4.12 Resumo

As redes de computador usam uma variedade de meios de transmissão, incluindo fios de cobre, fibras ópticas, transmissões de rádio e de microondas, infravermelho e feixes de laser. Cada meio de transmissão e tecnologia apresenta vantagens e custos. Por exemplo, embora um sistema infravermelho possa fornecer conexões de rede para computadores portáteis enquanto estes são movidos dentro de uma sala, um satélite em órbita pode ser necessário para fornecer a transmissão sem fio através de um oceano.

Exercícios

- 4.1 Quão forte é um cabo coaxial? Encontre um pedaço de cabo não-utilizado e remova a isolação da extremidade. A força do protetor lhe surpreende?
- 4.2 Investigue as conexões de rede usadas em sua localidade. Que tipos de meios são usados?
- 4.3 Que meios podem ser usados para uma conexão de rede que passe perto de um motor elétrico poderoso (por exemplo, um condicionador de ar)?
- 4.4 Alguns serviços de rede comerciais oferecem conexões de rede por satélite a algumas pessoas. A cada assinante é dada uma antena de prato pequena, usada para receber dados; ao assinante é dado também um modem, usado para enviar dados. Descubra por que um assinante não pode enviar dados ao satélite. Dica: leia sobre o tamanho das antenas nas estações terrestres que transmitem para um satélite.
- 4.5 Quão flexível é a fibra óptica? Para descobrir, veja se uma fibra quebra quando dobrada em torno de um arco com raio de 25cm, de 5cm, ou de 1cm.
- 4.6 Cabos de par trançado de categoria de número elevado são capazes de suportar taxas de dados mais elevadas, com a categoria 5 sendo o padrão atual para alta velocidade. Descubra se sua localidade usa o par trançado da categoria 5.
- 4.7 Se um satélite orbita a exatamente 20.000 milhas acima da superfície da Terra, quanto tempo leva para um sinal de rádio alcançá-lo e ser enviado de volta? E se o satélite estiver a exatamente 22.236 milhas? (Considere que o sinal se propaga na velocidade da luz e que o satélite leva 53 microsegundos para retransmitir um sinal.)

Sumário do Capítulo

- 5.1 Introdução 79**
- 5.2 A Necessidade de Comunicação Assíncrona 79**
- 5.3 Usando Corrente Elétrica para Enviar Bits 80**
- 5.4 Padrões de Comunicação 80**
- 5.5 Taxa de Baud, Enquadramento e Erros 82**
- 5.6 Comunicação Assíncrona Full Duplex 83**
- 5.7 Limitações do Hardware Real 84**
- 5.8 Largura de Banda do Hardware e a Transmissão de Bits 85**
- 5.9 O Efeito do Ruído na Comunicação 85**
- 5.10 Importância para a Comunicação de Dados 86**
- 5.11 Resumo 86**

Comunicação Local Assíncrona (RS-232)

5.1 Introdução

Como são dispositivos digitais, os computadores usam dígitos binários (bits) para representar dados. Assim, transmitir dados através de uma rede de um computador a outro significa enviar bits através do meio de transmissão subjacente. Fisicamente, os sistemas de comunicação usam corrente elétrica, ondas de rádio ou luz para transferir informações. Este capítulo explica como uma dessas formas, a corrente elétrica, pode ser usada para transferir informações digitais em pequenas distâncias. Ele mostra como os bits podem ser codificados e discute o mecanismo usado pela maioria dos computadores para enviar caracteres entre o teclado e o computador. O capítulo seguinte explica porque o mecanismo aqui descrito não pode ser usado para distâncias longas e descreve como a comunicação de longa distância é implementada.

Além de discutir a transmissão básica, este capítulo introduz as duas propriedades primárias de uma rede que podem ser medidas quantitativamente: largura de banda e atraso. Discute a motivação para usar medidas quantitativas e explica o relacionamento entre a largura de banda e a capacidade da rede. Os capítulos posteriores explicam como medidas similares podem ser aplicadas a sistemas de rede completos.

5.2 A necessidade de comunicação assíncrona

No sentido mais amplo do termo, uma comunicação é dita *assíncrona* se um remetente e um receptor não necessitam de coordenação antes que os dados possam ser transmitidos (isto é, o remetente e o receptor não sincronizam antes de cada transmissão). Assim, ao usar comunicação assíncrona, um remetente pode esperar por um tempo arbitrariamente longo entre as transmissões e transmitir sempre que existir dados para tanto. Em um sistema assíncrono, o receptor deve estar pronto para aceitar dados sempre que eles chegam. O assincronismo é especialmente útil para dispositivos como teclados, em que os dados são gerados quando um ser humano toca em uma tecla, e nenhum dado flui se o teclado está inativo.

De forma mais técnica, o hardware de comunicação é classificado como assíncrono se o sinal elétrico que o transmissor enviar não contiver informações que o receptor possa usar para determinar onde os bits individuais começam e terminam. Em vez disso, o hardware receptor deve ser cons-

truído para aceitar e interpretar o sinal que o hardware remetente gera. Este capítulo examinará o assincronismo que permite que um remetente transmita a qualquer momento e que requer que o receptor interprete o sinal; o capítulo seguinte descreverá o hardware síncrono de comunicação.

5.3 Usando corrente elétrica para enviar bits

Os sistemas eletrônicos de comunicação mais simples usam uma corrente elétrica pequena para codificar dados. Para compreender como a eletricidade pode codificar bits, imagine um fio que conecte dois dispositivos eletrônicos. Poderia ser usada tensão negativa para representar um 1 e tensão positiva para representar um 0. Por exemplo, para transmitir um bit 1, o dispositivo remetente coloca uma tensão positiva no fio por um curto período e então retorna o fio a zero volts. O dispositivo receptor detecta a tensão positiva e grava que um zero chegou. Similarmente, para enviar um bit 0, o dispositivo remetente coloca uma tensão negativa no fio por um curto período e retorna o fio ao zero volts. A Figura 5.1 ilustra como a tensão em um fio poderia variar sobre o tempo enquanto um dispositivo remetente transmite uma seqüência de bits.

A linha desenhada na Figura 5.1 é chamada de *diagrama de forma de onda*. Tais diagramas fornecem uma representação visual de como um sinal elétrico varia no tempo. O diagrama, por exemplo, mostra que um período mais longo decorreu entre a transmissão do quarto e do quinto bit do que entre outros.

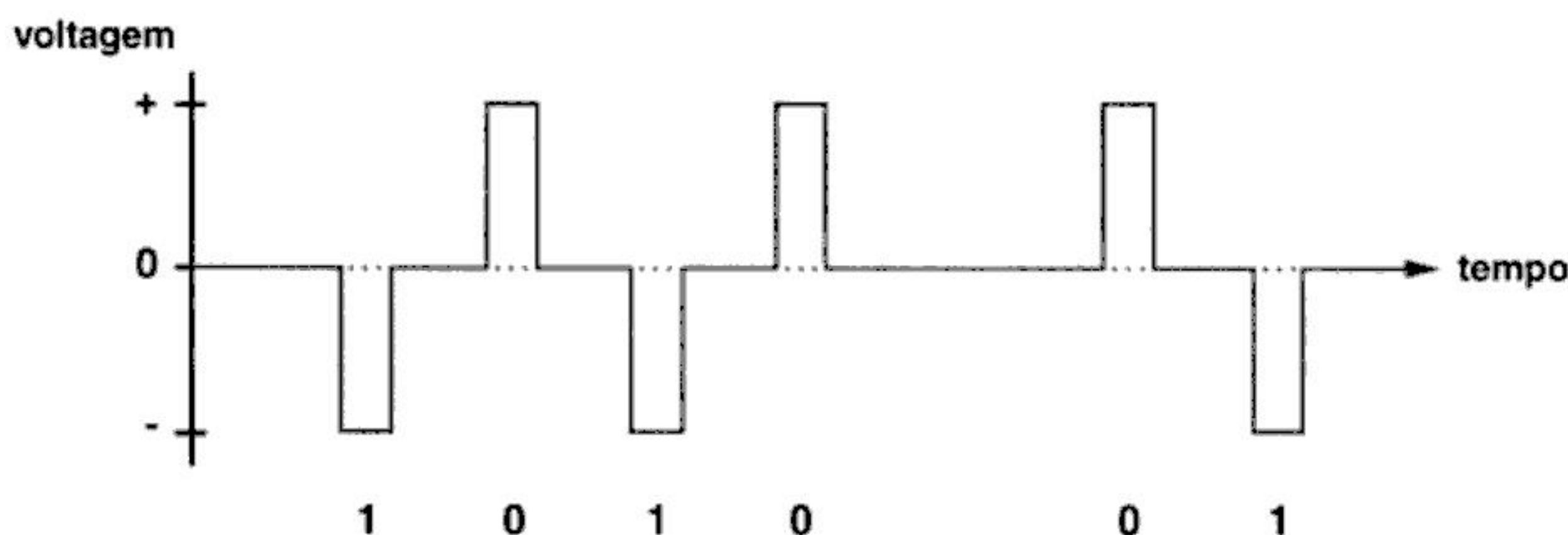


Figura 5.1 Ilustração de como as tensões positiva e negativa podem ser usadas para transmitir bits através de um fio. Neste exemplo, o remetente aplica uma tensão negativa para enviar um bit 1 ou uma tensão positiva para enviar um bit 0.

5.4 Padrões de comunicação

O exemplo na Figura 5.1 mostra uma maneira possível de usar tensão para transmissão digital. Entretanto, diversas perguntas permanecem sem resposta. Por exemplo, por quanto tempo o remetente deve manter uma tensão no fio para um único bit? Embora o remetente deva esperar o suficiente para que o hardware receptor detecte a tensão, esperar mais tempo do que o necessário desperdiça tempo. Qual é a taxa máxima em que o hardware pode mudar a tensão? Como um cliente pode saber se o hardware transmissor comprado de um vendedor trabalhará corretamente com o hardware receptor comprado de outro? Há uma forma de enviar mais dados na mesma quantidade de tempo?

Para assegurar-se de que os hardwares de comunicação construídos por vendedores diferentes interoperem, as especificações para sistemas de comunicação são padronizadas. Organizações como a União Internacional de Telecomunicações (*International Telecommunications Union, ITU*), a As-



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

O hardware RS-232 pode usar erros de enquadramento. Em particular, os teclados ASCII incluem freqüentemente uma tecla *BREAK*. A tecla *BREAK* não gera um caractere ASCII. Em vez disso, quando um usuário a pressiona, o teclado coloca a conexão de saída em estado 0 por muito mais tempo do que leva para enviar um único caractere (por exemplo, 2 segundos). Quando detecta que a linha se moveu para o estado 0, o receptor assume que um caractere começou a chegar e começa a extraír bits individuais. Entretanto, depois que todos os bits do caractere chegaram, o receptor espera a linha retornar ao estado 1 (isto é, espera um bit de parada). Se não encontrar um bit de parada como esperado, o receptor relata um erro de enquadramento, que pode ser usado pelo sistema receptor. Por exemplo, algumas aplicações usam a tecla *BREAK* como uma maneira de abortar uma aplicação – sempre que um usuário pressiona *BREAK*, o sistema relata um erro de enquadramento à aplicação que está usando o teclado. A aplicação interpreta o erro como uma requisição para abortar.

5.6 Comunicação assíncrona full duplex

Embora tenhamos descrito corrente elétrica fluindo através de um único fio, todos os circuitos elétricos requerem um mínimo de dois fios – a corrente flui para longe em um fio e volta em outro. O segundo fio é chamado freqüentemente de *terra*. Assim, quando o RS-232 é usado com par trançado, um dos fios carrega o sinal e o outro é um terra que fornece um caminho de retorno. Similarmente, quando um sinal é enviado ao longo de um cabo coaxial, o sinal viaja pelo centro do condutor e o protetor fornece o trajeto de retorno.

Em muitas aplicações de RS-232, os dados devem fluir nas duas direções ao mesmo tempo. Por exemplo, quando o RS-232 é usado para conectar um terminal ASCII a um computador, os caracteres viajam do teclado ao computador ao mesmo tempo em que viajam do computador à tela do terminal. A transferência simultânea nas duas direções é conhecida como *transmissão full duplex*, em contraste com a transferência em uma única direção, conhecida como *transmissão half duplex* ou *simplex*, a qual permite o fluxo de dados nas duas direções, mas somente em uma direção de cada vez. Para acomodar a transmissão *full duplex*, o RS-232 requer um fio para dados viajando em uma direção, um fio para dados viajando na direção reversa e um único fio terra usado para completar o caminho elétrico em ambas as direções. Além dos cabos de dados, o RS-232 padrão define um conjunto separado de cabos de controle e especifica como o hardware utiliza cada um dos cabos quando enviando dados⁵. Por exemplo, enquanto permanece capaz de receber caracteres, um receptor coloca uma tensão em um dos fios de controle, que o remetente interpreta como *autorização para enviar* (*clear to send*, ou ainda *CTS*). Para reduzir custos, o hardware RS-232 pode ser configurado de forma a operar sem os fios de controle (cada extremidade presume que a outra está funcionando). As conexões *full duplex* que ignoram sinais de controle são chamadas freqüentemente de *circuitos de três fios* (*three wire circuits*), porque necessitam de três fios para carregar dados (dois para os sinais que viajam em cada direção e um terra comum para o retorno). A Figura 5.3 ilustra um circuito de três fios.

Como mostra a figura, o fio terra conecta diretamente da terra em um dispositivo à terra no outro. Entretanto, os outros dois fios se cruzam: o fio conectado ao transmissor em um dispositivo se conecta ao receptor no outro. Para tornar os cabos mais simples, os projetistas decidiram que os computadores e os modems deveriam usar pinos opostos no conector padrão de 25 pinos – um computador transmite no pino 2 e recebe no pino 3, enquanto que um modem transmite no pino 3 e recebe no pino 2 (o fio terra usa o pino 7). Tecnicamente falando, os dois tipos de conectores são associados com *Equipamento de Comunicação de Dados* (*Data Communication Equipment, DCE*) e o *Equipamento Terminal de Dados* (*Data Terminal Equipment, DTE*). Portanto, um cabo que conecta um computador a

⁵ Conectores físicos usados com o RS-232 possuem 25, 15 ou 9 pinos, e são conhecidos pelos nomes de DB-25, DB-15 ou DB-9.

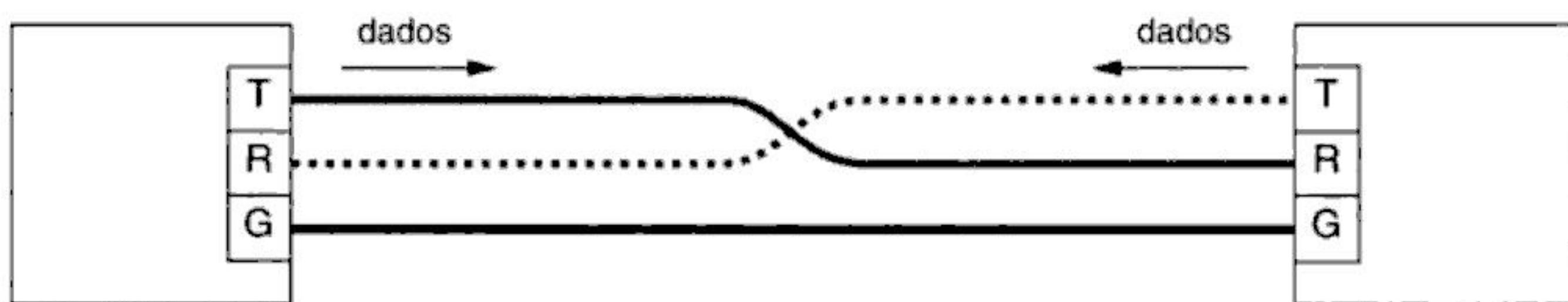


Figura 5.3 O cabeamento mínimo necessário para uma comunicação RS-232 full-duplex em que os fios de controle são omitidos. Embora os dois circuitos carreguem dados independentemente, eles podem compartilhar um único fio terra.

um modem tem um fio do pino 2 ao pino 2 e um fio do pino 3 ao pino 3. Entretanto, um cabo usado para conectar dois computadores (isto é, dois dispositivos de CTE) deve ter um fio do pino 2 ao pino 3 e um fio do pino 3 ao pino 2. Essa troca é freqüentemente chamada de *troca 2-3*.⁶

5.7 Limitações do hardware real

Quão rapidamente o hardware pode transmitir bits através de um fio? A ilustração na Figura 5.2 é um caso idealizado. Na prática, nenhum dispositivo eletrônico pode produzir uma tensão exata ou mudá-la de uma tensão a outra instantaneamente. Além disso, nenhum fio conduz eletricidade perfeitamente – quando flui corrente elétrica através dele, o sinal perde energia. Como resultado, leva um pequeno tempo para a tensão subir ou descer, e o sinal recebido não é perfeito. Por exemplo, a Figura 5.4 ilustra como um bit poderia parecer se as mudanças de tensão em uma linha de comunicação real fossem traçadas.

Como a maioria das tecnologias de comunicação, o RS-232 reconhece que o hardware real é imperfeito. O padrão especifica quão próximo do formato perfeito de onda um transmissor deve emitir, e quão tolerante à imperfeição um receptor deve ser. Por exemplo, o padrão não especifica que

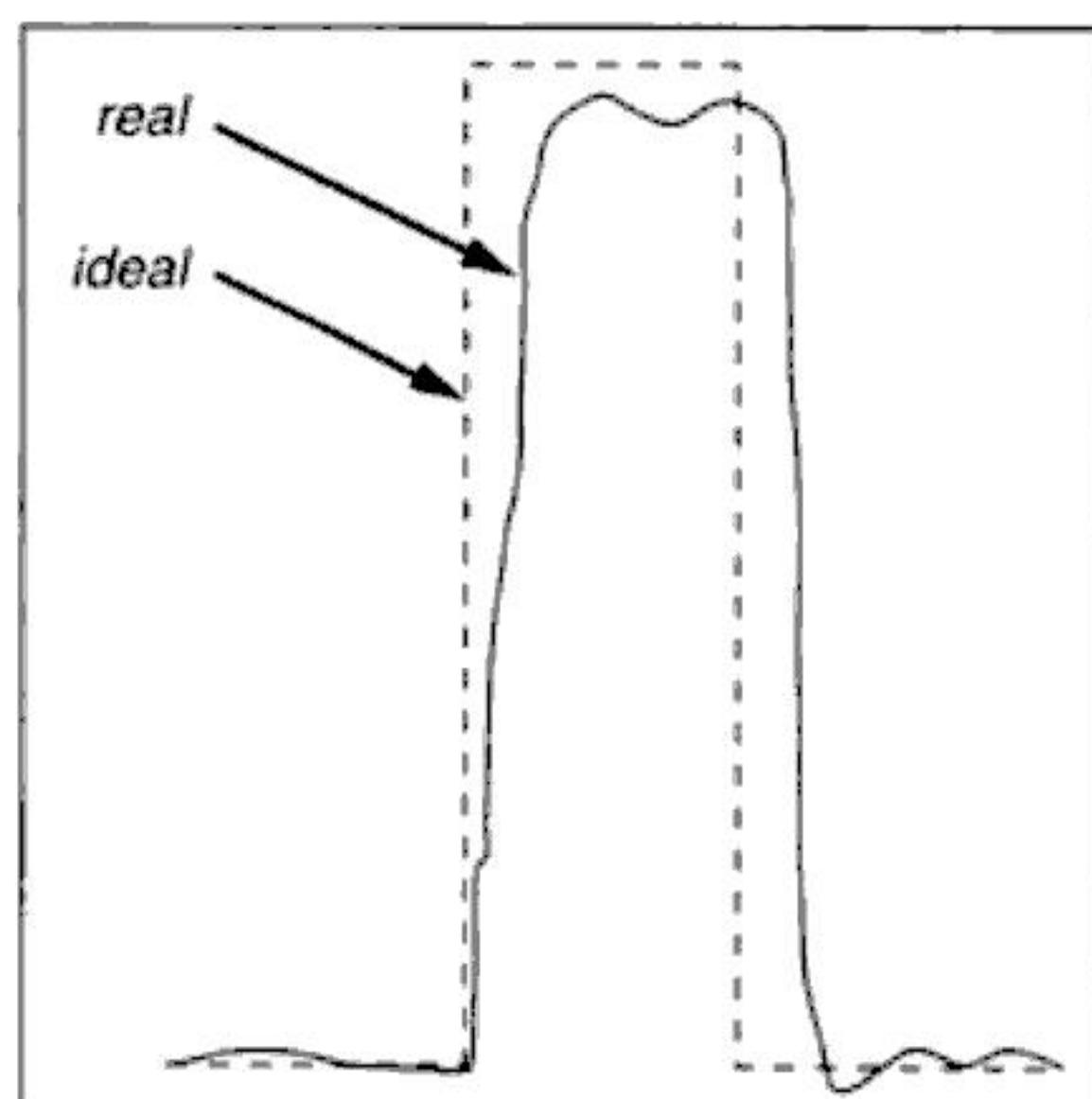


Figura 5.4 Uma ilustração da tensão emitida por um dispositivo real na transmissão de um bit. Na prática, as tensões são freqüentemente piores do que neste exemplo.

⁶ Um cabo que contém uma troca 2-3 é chamado de *null modem*; o capítulo seguinte explica o conceito de um modem.

um receptor deve medir a tensão exatamente no começo de cada bit. Em vez disso, recomenda a tomada de amostras durante o tempo alocado ao bit. Assim, um receptor aceitará sinais como ilustrado na Figura 5.4.

5.8 Largura de banda do hardware e a transmissão de bits

Saber que o hardware real não pode mudar tensões instantaneamente explica uma propriedade fundamental de sistemas da transmissão que está relacionada à velocidade em que os bits podem ser enviados. Cada sistema da transmissão tem uma *largura de banda (bandwidth)* limitada, que é a taxa máxima em que o hardware pode realizar mudanças de sinal. Se um remetente tentar transmitir mudanças mais rapidamente do que a largura de banda, o hardware não poderá prosseguir, pois não terá tempo suficiente para terminar uma mudança antes que o remetente tente fazer outra. Assim, algumas das mudanças serão perdidas.

A largura de banda é medida em *ciclos por segundo ou Hertz (Hz)*. É mais fácil pensar na largura de banda como o sinal de oscilação contínuo mais rápido que pode ser enviado através do hardware. Por exemplo, se um sistema da transmissão tiver uma largura de banda de 4000 Hz, então o hardware subjacente nesse sistema pode transmitir qualquer sinal que oscile para a frente e para trás em uma taxa igual ou menor a 4000 ciclos por segundo. Observe que cada sistema físico da transmissão tem uma largura de banda finita, porque as limitações da largura de banda derivam das propriedades físicas da matéria e da energia. Assim, qualquer sistema de transmissão que usar ondas de rádio, som, luz ou corrente elétrica terão uma largura de banda limitada⁷.

Na década de 20, um pesquisador descobriu a relação fundamental entre a largura de banda de um sistema de transmissão e do número máximo de bits por segundo que podem ser transferidos sobre esse sistema. Conhecido como *Teorema da Amostragem de Nyquist*, a relação fornece um limite teórico na velocidade máxima em que os dados podem ser enviados. Para um esquema da transmissão de dados como o RS-232, que usa dois valores de tensão para codificar dados, o Teorema de Nyquist estabelece que a taxa de dados máxima em bits por segundo que pode ser atingida sobre um sistema da transmissão da largura de banda B é $2B$. De maneira geral, se o sistema da transmissão usa K valores de tensão possíveis em vez de dois, o Teorema de Nyquist indica que a taxa de dados máxima em bits por segundo, D , é:

$$D = 2B \log_2 K$$

5.9 O efeito do ruído na comunicação

O Teorema de Nyquist fornece um máximo absoluto que não pode ser conseguido na prática. Em particular, os engenheiros observaram que um sistema real de comunicação está sujeito a quantidades pequenas de interferência de fundo, denominadas *ruído*, e que tal ruído torna impossível atingir a taxa máxima teórica de transmissão. Em 1948, Claude Shannon estendeu o trabalho de Nyquist para especificar a taxa de dados máxima que poderia ser conseguida sobre um sistema de transmissão que introduzisse ruído. O resultado, chamado de *Teorema de Shannon*⁸, pode ser expresso como:

$$C = B \log_2(1 + S/N)$$

onde C é o limite efetivo da capacidade do canal em bits por segundo, B é a largura de banda do hardware, S é a potência média do sinal e N é a potência média do ruído.

⁷ De fato, os sistemas biológicos têm limites da largura de banda também. Por exemplo, os cães podem ouvir sons que vão além da limitação de largura de banda dos ouvidos humanos.

⁸ O resultado também é chamado de Lei de Shannon-Hartley.

Geralmente, a S/N , conhecida como a relação *taxa sinal-para-ruído (signal-to-noise rate)*, não é representada diretamente. Em vez disso, os engenheiros citam a quantidade $10 \log_{10} S/N$, que é medida em decibéis (abreviado dB). Por exemplo, uma relação de S/N igual a 100 é 20 dB , e a relação de 1000 é 30 dB .

5.10 Importância para a comunicação de dados

Os Teoremas de Nyquist e de Shannon têm consequências para os engenheiros que projetam redes. O trabalho de Nyquist forneceu um incentivo para explorar maneiras complexas de codificar bits em sinais:

O Teorema de Nyquist incentiva os engenheiros a explorar algumas maneiras para codificar bits em um sinal porque uma codificação inteligente permite que mais bits sejam transmitidos na mesma unidade de tempo.

De certa forma, o Teorema de Shannon é mais fundamental porque representa uma limitação absoluta derivada das leis da física. Muito do ruído em uma linha de transmissão, por exemplo, pode ser atribuído à termodinâmica. Portanto,

O Teorema de Shannon informa aos engenheiros que não há codificação inteligente que supere as leis da física, colocando um limite fundamental no número de bits por segundo que pode ser transmitido em um sistema de comunicação real.

Na prática, o Teorema de Shannon ajuda a explicar quão rapidamente uma pessoa pode enviar dados através de uma chamada telefônica de voz. O sistema telefônico de voz tem uma relação sinal-para-ruído de aproximadamente 30 dB e uma largura de banda de aproximadamente 3000 Hz. Assim, de acordo com o Teorema de Shannon, o número máximo de bits por segundo que pode ser transmitido através de tal sistema é limitado a:

$$C = 3000 \log_2(1 + 1000)$$

ou aproximadamente 30.000 bps. Os engenheiros reconhecem isso como um limite fundamental – velocidades de transmissão mais rápidas somente serão possíveis depois que a relação sinal-para-ruído for melhorada.

Como os modems de conexão discada podem alcançar maior desempenho do que o teorema de Shannon permite? Uma possibilidade é a compactação – os dados são compactados antes da transmissão e descompactados depois da recepção. Claro, a compactação só funciona se os dados estiverem codificados de forma ineficiente. Por exemplo, quando a codificação de ASCII de 8 bits é usada para transferir e-mails que contenham apenas letras maiúsculas e minúsculas do alfabeto inglês e dígitos, apenas 62 das 256 possibilidades de 8 bits são usadas. Se os mesmos dados são codificados em caracteres de 6 bits, serão necessários 25% bits a menos. Deveria ser óbvio, porém, que embora a compressão reduza os números de bits requeridos para representar os dados, o teorema de Shannon ainda limita a velocidade na qual os dados comprimidos podem ser transmitidos.

5.11 Resumo

No sentido mais amplo, a comunicação assíncrona permite que um remetente transmita dados a qualquer momento e espere por um tempo arbitrariamente longo antes de transmitir outra vez. De forma mais técnica, o hardware de comunicação é assíncrono se o sinal elétrico não contiver informações que um receptor possa usar para encontrar o começo e o fim dos bits. Criado originalmente para definir a interação entre um computador e um modem, o padrão RS-232 transformou-se no

padrão mais amplamente aceito para a transmissão assíncrona de caracteres sobre pequenas distâncias. O RS-232 é usado para comunicação entre um teclado e um computador, assim como para comunicação através das portas seriais de um computador.

De acordo com o padrão RS-232, um transmissor deve deixar uma tensão negativa na linha de comunicação quando não tem dados a enviar. O transmissor precede cada caractere com um bit de começo e termina cada caractere com um bit de parada. O bit de começo informa ao receptor que um caractere está chegando, e o de parada permite que um receptor detecte que todos os bits do caractere chegaram no tempo alocado.

Já que qualquer sistema físico usado para comunicação tem limites na velocidade em que pode mudar o estado, sistemas físicos não podem transmitir bits em uma velocidade arbitrariamente rápida. Por exemplo, componentes eletrônicos não podem mudar a tensão em um fio instantaneamente. A velocidade em que o hardware pode mudar o estado é conhecida como a largura de banda do hardware; a largura de banda de um sistema de transmissão pode ser medida. Os pesquisadores descobriram duas relações fundamentais. O Teorema de Nyquist define a relação entre a largura de banda do hardware e a taxa máxima teórica em que os dados podem ser enviados. O Teorema de Shannon estabelece um limite na taxa em que os dados podem ser enviados na presença de ruído.

Para estudos futuros

Detalhes adicionais sobre conexões de modem nulas (null modem) para RS-232 e os fios utilizados com conectores DB-9 podem ser encontrados em:

<http://www.nullmodem.com/DB-9.htm>

Exercícios

- 5.1 O termo *baud* advém do nome Emile Baudot. Descubra quem era Baudot e que contribuição fez aos sistemas de comunicação.
- 5.2 Desenhe o diagrama do formato de onda que resulta quando a palavra *bit* é enviada em ASCII através de uma conexão RS-232. Dica: o Apêndice 2 contém os códigos de caractere de 7 bits ASCII.
- 5.3 Suponha que alguém enviou 10000 caracteres de 7 bits através de uma conexão RS-232 que operava a 9600 baud. Quanto tempo seria necessário para a transmissão? (Dica: lembre-se de contabilizar um bit de começo e um bit de parada em cada caractere.)
- 5.4 Use o Teorema de Nyquist para determinar a taxa máxima de bits por segundo em que podem ser enviados dados através de um sistema de transmissão que tenha uma largura de banda de 4000 Hz e use quatro valores de tensão para codificar informações.
- 5.5 Leia sobre RS-232. Que finalidade é atribuída a cada um dos 25 fios em um conector DB-25?
- 5.6 As portas seriais em alguns computadores usam conectores com menos de 25 pinos. Leia sobre o conector DB-9 usado para portas seriais em PCs. Quando o RS-232 é usado com um conector DB-9, que sinais são omitidos?
- 5.7 Leia sobre o hardware RS-232. Quantas vezes o hardware mede cada bit?
- 5.8 Estenda o exercício anterior calculando quão próximas as taxas do hardware devem estar nas interfaces RS-232 para permitir a transferência bem-sucedida de dados. Se o hardware RS-232 do remetente produz bits que são cinco por cento menores do que o hardware RS-232 do receptor espera, o receptor aceitará os caracteres resultantes?
- 5.9 O que acontece se um transmissor RS-232 e um receptor forem programados para enviar e receber em taxas em baud diferentes? Para descobrir, conecte uma linha serial entre dois computadores e faça as taxas desiguais (você não danificará o hardware).
- 5.10 A maioria dos hardwares RS-232 permite ao computador especificar a taxa de dados e o número de bits de parada a serem utilizados. Se um transmissor estiver programado para usar dois bits de parada, mas um receptor estiver programado para exigir somente um, os dados serão recebidos corretamente? Se assim for, qual é a desvantagem de usar um bit extra de parada?



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

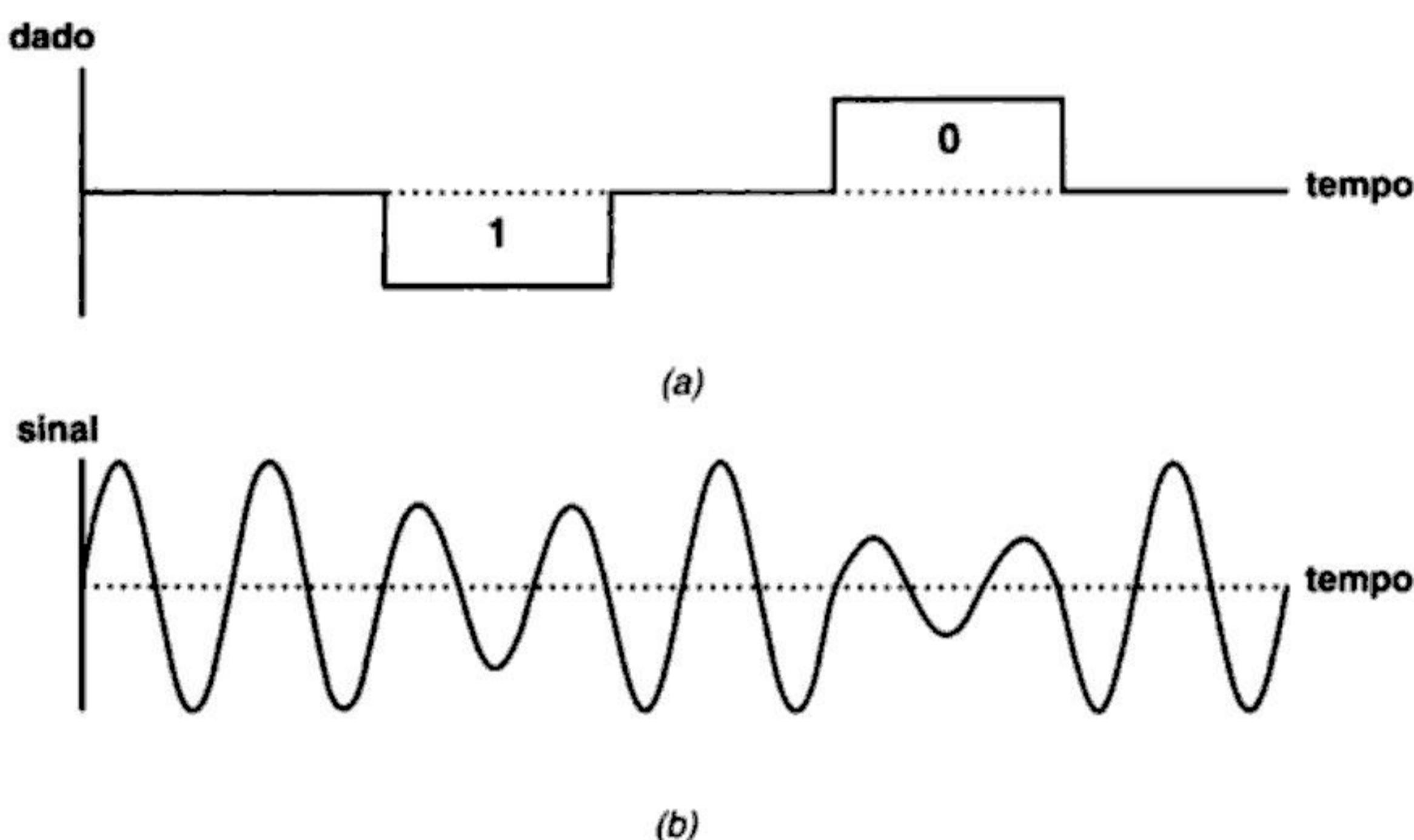


Figura 6.2 (a) Um sinal digital e (b) a onda que resulta da modulação da amplitude usando o sinal em (a). A portadora é reduzida a 2/3 da potência total para codificar um bit 1 e 1/3 da potência para codificar um bit 0.

observe na Figura 6.3 que as seções horizontais de uma onda senoidal comum foram removidas, as partes restantes, agrupadas e as quebras, juntadas com linhas verticais nos pontos indicados por setas. O tamanho da seção removida determina a quantidade de deslocamento.

Foi dito que a taxa em baud de um sistema de transmissão é o número de mudanças que o hardware pode fazer por segundo. A vantagem principal de mecanismos como a modulação de deslocamento de fase surge da sua habilidade de codificar mais de um bit em uma dada mudança. Por exemplo, a Figura 6.3 ilustra como a fase pode ser deslocada em quantidades diferentes.

Um ciclo completo da onda portadora consiste em um arco positivo seguido por um arco negativo. Na Figura 6.3, cada um dos dois primeiros deslocamentos salta metade de um ciclo completo, enquanto o terceiro salta três quartos de um ciclo². Geralmente, os deslocamentos de fase são esco-

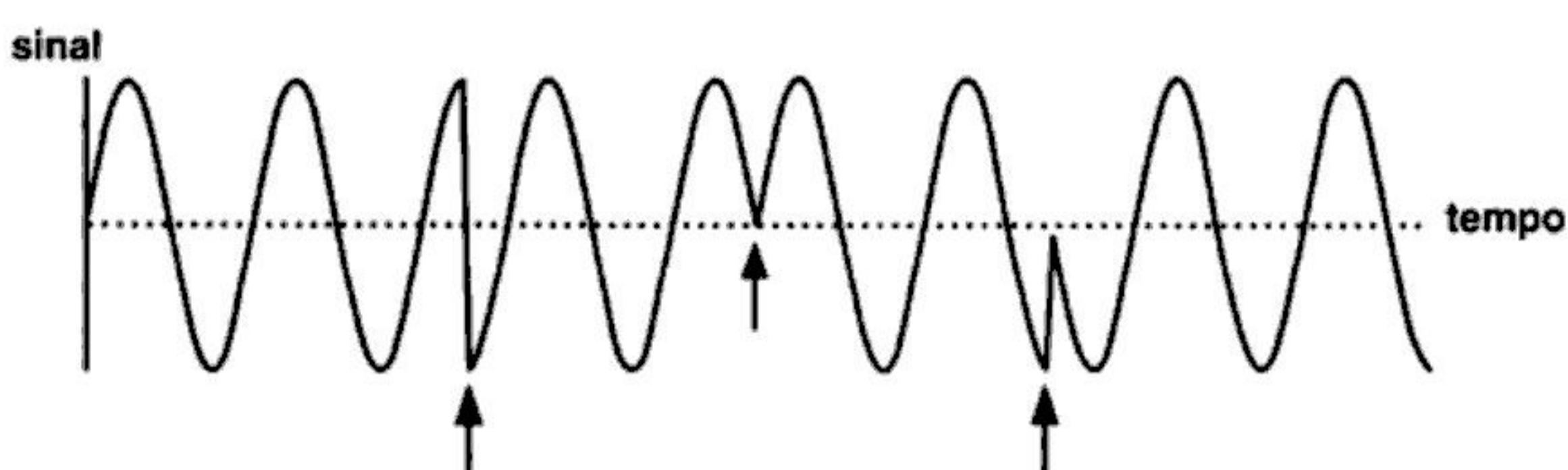


Figura 6.3 Uma ilustração da modulação por deslocamento de fase. As setas indicam os pontos em que a portadora salta abruptamente para um novo ponto no ciclo.

² Matematicamente, uma senóide completa um ciclo em 2π radianos. Portanto, os dois primeiros deslocamentos na Figura 6.3 são π radianos, cada um, e o terceiro é $3\pi/2$ radianos.

lhidos de modo que cada um represente uma potência de dois valores possíveis. O remetente pode, então, usar bits de dados para selecionar o deslocamento. Em um sistema que pode deslocar a fase por oito quantidades possíveis (isto é, 2^3), um transmissor usa três bits de dados para selecionar qual dos oito valores de deslocamento será usado. O receptor determina quanto a portadora foi deslocada e usa o deslocamento para recriar os bits que produziram a mudança. Isto é, se um transmissor usa T bits para criar um deslocamento de fase, o receptor pode extrair todos os T bits observando a quantidade de deslocamento. Uma vez que cada deslocamento codifica T bits, a taxa máxima de dados que pode ser enviada usando modulação por deslocamento de fase é $2B\log_2 2^T$, ou $2BT$, onde B é o número de mudanças do sinal por segundo. De acordo com a definição dada, B é a taxa de baud do hardware. Assim, ao usar a modulação por deslocamento de fase, o número de bits por segundo que o sistema pode transferir é um múltiplo da taxa em baud.

6.3 Hardware do modem usado para a modulação e a demodulação

Um circuito de hardware que aceite uma seqüência de bits de dados e aplique modulação a uma onda portadora de acordo com os bits é chamado de *modulador*; um circuito de hardware que aceite uma onda modulada da portadora e recrie a seqüência de bits dos dados que foi usada para modular a portadora é chamado de *demodulador*. Assim, a transmissão dos dados através de uma longa distância requer um modulador em uma extremidade da linha de transmissão e de um demodulador na outra.

Na prática, a maioria dos sistemas de rede são full duplex (isto é, permitem que os dados fluam em ambas as direções). Para suportar tal comunicação, cada posição necessita de um modulador, usado para transmitir dados, e de um demodulador, usado para receber dados. Para manter os custos baixos e tornar o par de dispositivos fácil de instalar e operar, os fabricantes combinam os circuitos em um único dispositivo chamado *modem* (*modulador* e *demodulador*). A Figura 6.4 ilustra como um par de modems pode ser usado para conectar dois computadores através de uma longa distância.

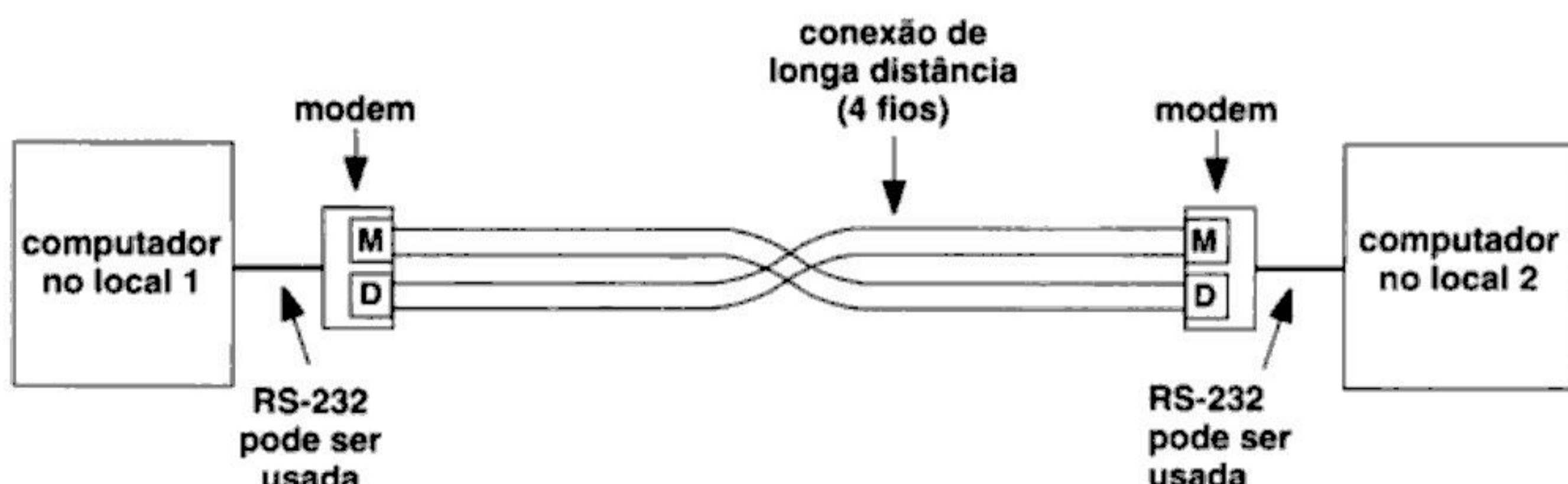


Figura 6.4 O uso de dois modems para comunicação de longa distância através de um circuito de 4 fios. O modulador em um modem conecta-se ao demodulador no outro. Um par de fios é necessário para cada conexão.

6.4 Circuitos alugados de dados analógicos (leased analog data circuits)

Muitas empresas usam um ou mais circuitos de 4 fios como parte de sua rede de transmissão de dados. Quando o circuito conecta duas posições em um único local, a própria empresa pode instalar os fios necessários. As empresas privadas, no entanto, não podem instalar circuitos através de longas distâncias, porque algumas normas do governo permitem que somente as empresas de ser-

viço público passem fios através da propriedade pública (por exemplo, através de uma rua). Felizmente, a fiação necessária pode ser obtida de uma companhia telefônica. Em particular, as empresas telefônicas permitem que as empresas aluguem um circuito entre duas posições quaisquer.

Para compreender por que a companhia telefônica pode fornecer a fiação, é necessário conhecer dois fatos. Primeiramente, quando uma companhia telefônica instala cabos, cada um inclui fios extras que podem ser usados para expansão futura. Assim, os cabos telefônicos já existentes contêm fios que não estão sendo usados. Em segundo lugar, embora não venda os fios nos cabos, uma companhia telefônica concorda em alugar os fios por uma taxa mensal (a taxa depende da distância abrangida e da largura de banda dos fios). Um circuito alugado consiste geralmente em quatro fios que não se conectam de nenhuma maneira ao sistema telefônico do dial-up – os fios podem ser usados apenas com modems especiais como descrito acima. Como os bits viajam um de cada vez através desses circuitos, os profissionais usam os termos *círculo de dados serial*, *linha serial* ou *linha serial alugada* para descrever tal conexão.

Uma vez que uma conexão de uma companhia telefônica foi alugada, um modem deve ser instalado em cada extremidade para que a comunicação seja possível. Após isso, a linha alugada está disponível para o envio de dados. A principal vantagem de tal arranjo deriva de sua constante disponibilidade – podem ser enviados dados a qualquer momento, vinte e quatro horas por dia. As principais desvantagens são fruto da conectividade limitada e do custo – a linha alugada conecta apenas dois pontos e, quem quer que seja, o locatário deve pagar a taxa mensal mesmo que a linha não esteja sendo usada.

6.5 Freqüência óptica, de rádio e modems dial-up

Além dos fios exclusivos, os modems são usados também com outros meios, incluindo a transmissão em RF, fibras de vidro e conexões telefônicas convencionais. Por exemplo, um par de modems de *freqüência de rádio (RF)* pode ser usado para enviar dados usando um sinal de freqüência de rádio, ou um par de modems ópticos pode ser usado para enviar dados através de um par de fibras de vidro usando luz. Embora tais modems usem tecnologia inteiramente diferente do que os que operam sobre fios exclusivos, o princípio permanece o mesmo: na extremidade remota, um modem transforma dados em um sinal modulado; na extremidade receptora, os dados são extraídos do sinal modulado.

Os modems RF tornaram-se especialmente atrativos por causa do crescente interesse na *ligação em rede sem-fio (wireless networking)*. Um pequeno modem RF conectado a um notebook, por exemplo, possibilita que o computador seja movido por um edifício sem perder a conectividade de rede, da mesma forma que um telefone portátil pode ser movido. Modems RF mais poderosos possuem estabelecer um link de comunicação sem-fio sobre distâncias mais longas (por exemplo, quilômetros).

Outra aplicação interessante de modems envolve o sistema dial-up de telefone. Conforme ilustrado na Figura 6.5, um *modem dial-up* é conectado a uma linha telefônica comum.

Os modems dial-up diferem de três maneiras significativas dos modems de 4 fios previamente descritos. Além dos circuitos para envio de dados, um modem dial-up contém circuitos que imitam um telefone – pode simular o levantamento do aparelho, a discagem ou o desligar. E, como o sistema telefônico foi projetado para carregar som, um modem dial-up usa uma portadora um tom audível³. Portanto, um modem dial-up deve conter circuitos para enviar e receber áudio sobre a linha telefônica (além de enviar e receber uma portadora, um modem dial-up pode detectar um tom de discagem). Além disso, embora enviem todos os dados por um único canal de voz, um par de modems dial-up oferece comunicação full duplex. Isto é, uma única conexão telefônica entre dois mo-

³ A portadora usada pelo modem dial-up é o som ouvido se alguém acidentalmente levanta o fone do gancho enquanto um modem está usando a linha telefônica.



Figura 6.5 Ilustração dos modems dial-up que usam o sistema telefônico de voz para se comunicar. Para o sistema telefônico, um modem dial-up parece ser um telefone.

demis dial-up geralmente permite que os dados fluam em ambas as direções. Na prática, os modems devem usar tons diferentes de portadora ou se coordenarem para evitar que ambos transmitam ao mesmo tempo.

Os modems que coordenam o envio de dados são chamados de *modems half-duplex* ou *modems de 2 fios*, para distingui-los do tipo 4 fios descrito anteriormente. Para coordenar, o par de modems de 2 fios concorda em se alternar no envio de dados. Um modem envia dados e permite então que o outro modem envie. Tal coordenação ocorre automaticamente; um usuário desconhece que os modems estão se alternando.

Para usar um par de modems dial-up, um modem deve começar a operação esperando uma chamada telefônica em um número de telefone conhecido, T . Diz-se que o modem em espera está em *modo de resposta (answer mode)*. O outro modem começa em *modo de chamada*, com o número de telefone a discar. O modem que chama simula o levantar do telefone, espera por um tom de discagem e então disca. Quando o telefone está tocando, o modem que está na modalidade de resposta responde à chamada e envia uma onda portadora, que inicia a comunicação. O modem que chama detecta a portadora e responde enviando uma portadora própria. Uma vez que os dois modems concordam sobre as portadoras, podem ser enviados dados em uma ou outra direção modulando a portadora como descrito acima. Depois que uma aplicação termina de se comunicar através de um modem dial-up, a aplicação instrui o modem para terminar a chamada⁴.

Os computadores que usam modems não sabem sobre os meios subjacentes – um modem não informa ao computador se está usando fios, fibras ópticas, uma conexão telefônica dial-up ou outro meio. Uma aplicação que utilize um modem poderia ser capaz de deduzir os meios subjacentes medindo o atraso e a largura de banda do canal, mas um software de computador raramente tenta fazê-lo. Em vez disso, a maioria dos sistemas de computadores usa modems meramente como uma maneira de enviar bits através de longas distâncias.

Em resumo,

Um par de modems é necessário para comunicação de longa distância através de uma linha alugada; cada modem contém circuitos separados para enviar e receber dados digitais. Para enviar dados, um modem emite uma onda portadora contínua, que ele modula de acordo com os valores dos bits sendo transferidos. Para receber dados, um modem detecta a modulação na portadora recebida e usa-a para recriar os bits de dados.

⁴ Como uma alternativa para os modems de chamada discada padrão, um modem V.90 ou V.92 utiliza um esquema assimétrico no qual um ISP tem um conexão digital (ISDN) e um assinante tem uma conexão analógica padrão; a assimetria torna possível que o caminho da carga de caracteres (downstream) tenha maior desempenho.

6.6 Freqüências e multiplexação da portadora

As redes de computadores que usam uma onda modulada de portadora para transmitir dados são similares às estações de televisão que usam uma onda modulada de portadora para difundir vídeo. As similaridades fornecem a intuição necessária para entender um princípio fundamental:

Dois ou mais sinais que usam freqüências de portadora diferentes podem ser transmitidos sobre um único meio simultaneamente sem interferência.

Para entender o princípio, considere como a transmissão de televisão funciona. A cada estação de televisão é atribuído um número de canal em que se difunde um sinal. Na realidade, um número de canal é meramente um atalho para a freqüência em que a portadora da estação oscila. Para receber uma transmissão, um receptor de televisão deve ser ajustado à mesma freqüência que o transmissor. Mais importante, uma dada cidade pode conter muitas estações de televisão que transmitem em freqüências separadas simultaneamente. Um receptor seleciona uma freqüência para receber em um dado momento.

A televisão a cabo ilustra que o princípio se aplica a muitos sinais que viajam através de um fio. Embora um assinante de cabo tenha somente um fio físico que o conecta à companhia de cabo, o assinante recebe muitos canais de informação simultaneamente. O sinal de um canal não interfere no de outro, possibilitando assistir a um show no canal 6 sem receber nenhuma interferência dos sinais dos canais 5 ou 7.

As redes de computadores usam o princípio de canais separados para permitir que comunicações múltiplas compartilhem uma única conexão física. Cada remetente transmite um sinal usando uma freqüência particular de portadora. Um receptor configurado para aceitar uma portadora de uma dada freqüência não será afetado pelos sinais enviados em outras freqüências. Assim, as múltiplas portadoras podem passar sobre o mesmo fio ao mesmo tempo sem interferência.

A *Multiplexação por Divisão de Freqüência (Frequency Division Multiplexing, FDM)* é o termo técnico aplicado a um sistema de rede que utiliza freqüências múltiplas de portadora para permitir que sinais independentes viajem por um meio. A tecnologia FDM pode ser usada para enviar sinais pelo fio, RF ou fibra óptica. A Figura 6.6 ilustra o conceito e mostra os componentes de hardware necessários para FDM.

Na teoria, contanto que cada portadora opere em uma freqüência diferente das demais, ela permanece independente. Na prática, entretanto, duas portadoras operando em freqüências muito próximas ou em múltiplos exatos de outra freqüência podem interferir uma na outra. Para evitar problemas, os engenheiros que projetam sistemas de rede FDM escolhem uma separação mínima entre as portadoras⁵.

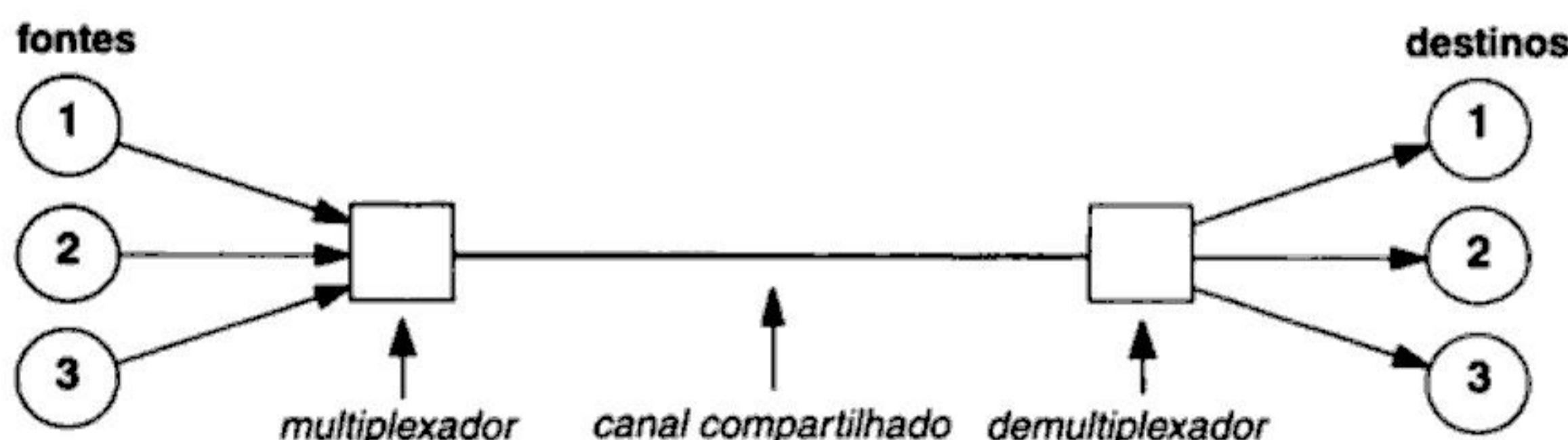


Figura 6.6 O conceito de multiplexação por divisão de freqüência. Cada par de fonte e destino pode enviar dados por canal compartilhado sem interferência. Na prática, cada extremidade requer um multiplexador e um demultiplexador para uma comunicação bidirecional, e um multiplexador pode necessitar de circuitos para gerar as ondas da portadora.

⁵ Os requisitos para uma separação mínima entre freqüências também se aplicam a estações de televisão e de rádio.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

Embora possam usar as mesmas técnicas de modulação que estações de rádio ou de televisão, a maioria dos modems usa técnicas como a modulação por deslocamento de fase, que funcionam bem para dados digitais. Para implementar a modulação por deslocamento de fase, o modem remetente muda a onda portadora saltando abruptamente para um ponto novo no ciclo. A vantagem principal da modulação por deslocamento de fase encontra-se em sua habilidade de codificar mais de um bit de dados em cada deslocamento.

O conceito de multiplexação é fundamental para as redes de computadores. A multiplexação permite que múltiplas fontes enviem a múltiplos destinos através de um canal de comunicação compartilhado. A multiplexação por divisão de freqüência funciona como a televisão a cabo – os múltiplos sinais podem viajar através de um fio simultaneamente porque cada um usa uma freqüência original da portadora. A multiplexação por divisão de tempo requer que os remetentes alternem a transmissão no meio compartilhado.

Exercícios

- 6.1 Considere o uso de modulação por amplitude com uma onda senoidal que opere em uma freqüência de 4000 Hz. Quantos bits por segundo podem ser codificados? Por quê? (Dica: são necessárias informações do Capítulo 4 para resolver este problema.)
- 6.2 Explique por que cada estação de rádio em uma área deve usar uma freqüência original de portadora.
- 6.3 Verifique com uma companhia local de telefonia quanto custa por ano alugar um circuito de dados de 4 fios que se estenda por uma milha.
- 6.4 Embora os modems dial-up necessitem de sofisticada eletrônica para lidar com os detalhes de discar um telefone, eles são usualmente muito mais baratos do que os modems equivalentes de 4 fios. Explique por quê. (Sugestão: pense na economia.)
- 6.5 Investigue os vendedores comerciais de modems para encontrar a taxa máxima de dados em bits por segundo disponível sobre um modem de telefone dial-up.
- 6.6 No exercício anterior, use o Teorema de Shannon para calcular a relação entre sinal e ruído necessária em uma linha telefônica dial-up para obter a taxa de dados máxima do modem. (Suponha uma largura de banda de 3000 Hz.)
- 6.7 O autor usa um multiplexador e um demultiplexador que permite que três fontes de dados de 8 bits separadas enviem a três destinos separados sobre um único canal. O canal pode enviar apenas caracteres de 8 bits, e qualquer fonte pode transferir qualquer valor possível de 8 bits a qualquer momento. Desenvolva um mecanismo de multiplexação que trabalhe com hardware de 8 bits para aceitar dados das três fontes e para enviá-los através de um canal compartilhado. (Dica: planeje uma maneira de dizer ao receptor qual fonte enviou cada item de dados.)
- 6.8 Escreva dois programas de computador que implementem o esquema de multiplexação que você projetou no exercício anterior. Faça um programa simular um multiplexador lendo caracteres de três arquivos e produzindo um quarto arquivo. Para garantir que os dados possam ser misturados, leia um caractere do primeiro arquivo, um do segundo, um do terceiro e assim por diante. Faça o segundo programa tomar a saída do primeiro e decodificá-la em três arquivos de saída (que deveriam ser idênticos aos arquivos de entrada).
- 6.9 Substituir um multiplexador que empregue Multiplexação por Divisão de Tempo (*Time Division Multiplexing – TDM*) por um que use Multiplexação por Divisão de Freqüência (*Frequency Division Multiplexing – FDM*) pode melhorar o throughput. Explique por quê.
- 6.10 Leia sobre pesquisa em *Multiplexação por Divisão de Onda* (*Wave Division Multiplexing*, WDM) para modems ópticos. Quantas freqüências de luz se consegue enviar simultaneamente através de uma fibra óptica com WDM?

Transmissão de Pacotes

***Pacotes, quadros, Redes Locais,
Redes de Longo Alcance, endereços
de hardware, bridges, switches,
roteamento e protocolos***

Sumário do Capítulo

- | | | | |
|--|-------------|---|-----|
| | 7.1 | Introdução | 101 |
| | 7.2 | O Conceito de Pacotes | 101 |
| | 7.3 | Pacotes e Multiplexação por Divisão de Tempo | 103 |
| | 7.4 | Pacotes e Quadros de Hardware | 103 |
| | 7.5 | Byte Stuffing | 104 |
| | 7.6 | Erros de Transmissão | 105 |
| | 7.7 | Bits de Paridade e Verificação de Paridade | 106 |
| | 7.8 | Probabilidade, Matemática e Detecção de Erros | 107 |
| | 7.9 | Detectando Erros com Checksums | 108 |
| | 7.10 | Detectando Erros com Verificação de Redundância Cíclica | 109 |
| | 7.11 | Combinando Blocos Básicos | 110 |
| | 7.12 | Erros de Rajadas | 111 |
| | 7.13 | Formato de Quadro e Mecanismos de Detecção de Erro | 111 |
| | 7.14 | Resumo | 112 |

Pacotes, Quadros e Detecção de Erro

7.1 Introdução

Os capítulos anteriores descrevem como os níveis mais baixos de hardware transmitem bits individuais por meios como fios de cobre ou fibras de vidro. Embora tais detalhes sejam interessantes, somente engenheiros que projetam hardware trabalham com bits individuais ou técnicas de modulação. A maioria das redes de computadores fornece uma interface mais conveniente, que permite a um computador enviar múltiplos bytes de dados através da rede sem manipular bits individuais e sem conhecer como o hardware subjacente codifica bits em sinais.

Este capítulo descreve uma idéia fundamental em redes de computadores. Discute o conceito de pacotes e explica como um remetente e um receptor se coordenam para transferir um pacote. Mostra também como os pacotes podem ser implementados em uma rede orientada a caractere usando um formato de quadro simples. Finalmente, o capítulo explica erros de transmissão e discute mecanismos que as redes usam para descobri-los.

Os capítulos posteriores expandem a noção de pacote e descrevem como tecnologias de redes particulares manipulam pacotes. Eles mostram mais exemplos e examinam alguns detalhes.

7.2 O conceito de pacotes

A maioria das redes de computadores não transfere dados como uma string arbitrária de bits contínuos. Em vez disso, o sistema de rede divide dados em blocos pequenos, chamados *pacotes*, que envia individualmente. As redes de computadores são freqüentemente chamadas de *redes de pacote* ou *redes de comutação de pacotes* porque usam tecnologia de pacote.

Dois fatos motivam o uso de pacotes. Primeiro, um remetente e um receptor precisam coordenar a transmissão para assegurar que os dados chegam corretamente. Será aprendido que, quando ocorrem erros de transmissão, os dados podem ser perdidos. A divisão dos dados em pequenos blocos ajuda o remetente e o receptor a determinar que blocos chegaram intactos e quais não chegaram. Segundo, como os circuitos de comunicação e o respectivo hardware de modem são caros, freqüentemente múltiplos computadores compartilham conexões e hardware subjacente. Para assegurar que todos os computadores recebem acesso justo e imediato a uma instalação de comunicação compartilhada, um sistema de rede não pode permitir que um computador impeça o acesso de ou-

tros. O uso de pacotes pequenos ajuda a assegurar justiça no acesso. Para entender como, considere a alternativa usada em redes de computadores primitivas.

As primeiras redes não garantiam acesso justo. Em vez disso, permitiam que um programa aplicativo segurasse um recurso de comunicação compartilhado por um tempo arbitrário – permitia-se que um aplicativo fosse até o fim antes que outro pudesse começar a usar o recurso. Para evitar que um computador detenha uma rede por um tempo arbitrário, as redes de computadores modernas obrigam o uso de pacotes. A rede permite que um computador envie um pacote e então impede que o computador envie novamente. Enquanto isso, permite que outro computador envie um pacote, e assim por diante. Um computador pode deter um recurso compartilhado por tempo suficiente apenas para enviar um único pacote e deve esperar até que outros computadores tenham vez antes de enviar um segundo pacote.

Para entender como o uso de pacotes permite serviço imediato, suponha que uma rede tenha concedido a um programa aplicativo o uso exclusivo de uma rede até que o aplicativo termine. Por exemplo, suponha que os quatro computadores na Figura 7.1 compartilhem um canal de comunicação e que usem o canal para transferir arquivos. Enquanto o computador *A* envia um arquivo para o computador *D*, os computadores *B* e *C* devem esperar.

Quanto tempo exige uma transferência de arquivos? Se um arquivo contiver 5 megabytes (um arquivo de dados típicos) e o sistema de comunicação puder transferir 56.000 bits por segundo (uma taxa típica de uma rede de longa distância), a transferência exigirá quase 12 minutos. O exemplo mostra claramente isto:

Qualquer sistema de rede que concede a um aplicativo o uso exclusivo de recursos compartilhados bloqueará outros computadores por períodos intoleravelmente longos.

Em contraste, considere os atrasos introduzidos quando a rede na Figura 6.1 exige que os computadores dividam dados em pacotes de 1000 bytes. Suponha que o computador *A* comece enviando dados para *D*. Suponha também que depois de *A* começar a enviar, o computador *B* precisa enviar dados para *C*. Após *A* enviar um pacote, a rede permitirá que *B* envie um também. Como um pacote contém apenas 8000 bits de dados e o hardware pode transferir dados a 56.000 bits por segundo, um pacote completo pode ser enviado em apenas 0,143 segundos. Conseqüentemente, *B* espera no máximo 143 ms antes de começar a enviar. Pode-se resumir:

Para permitir que remetente e receptor se coordenem e para assegurar que todos os computadores que compartilham um recurso de rede tenham acesso justo e imediato, a maioria das redes de computadores divide dados em pequenos blocos, chamados de pacotes. Os computadores se alternam enviando pacotes através do recurso compartilhado. Como cada pacote é pequeno, nenhum computador tem um atraso longo enquanto estiver aguardando sua vez de acessar.



Figura 7.1 Uma das razões por que as redes de computadores usam pacotes. Enquanto um par de computadores se comunica, os outros devem esperar.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

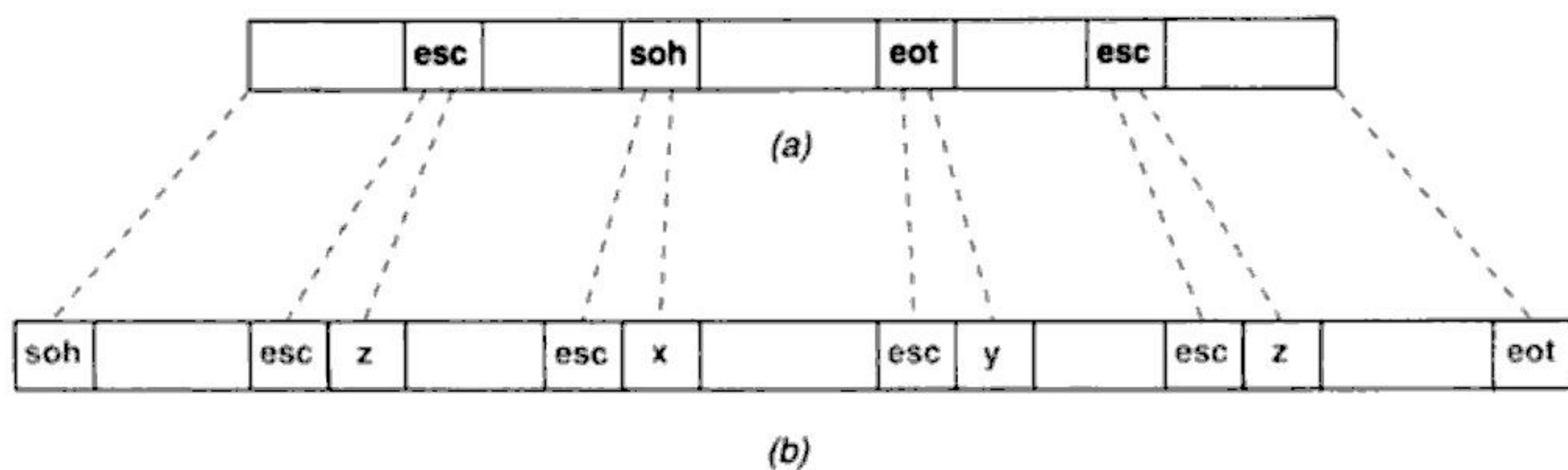


Figura 7.5 Ilustração de byte stuffing, onde (a) é um exemplo de dados que inclui caracteres como *soh*, e (b) é o quadro após o byte stuffing. As linhas tracejadas mostram as localizações nos dados originais onde os caracteres foram substituídos ou adicionados caracteres novos.

quêntemente, porém, a interferência muda somente o sinal usado para transmissão sem danificar o equipamento. Uma pequena mudança no sinal elétrico pode fazer com que o receptor interprete mal um ou mais bits de dados. De fato, a interferência pode destruir completamente um sinal, significando que, embora o remetente transmita, o receptor não detecta a chegada de dados. Surpreendentemente, a interferência em um circuito de transmissão completamente inativo pode criar o efeito oposto – embora o remetente nada tenha transmitido, um receptor poderia interpretar a interferência lida como uma sequência válida de bits ou caracteres. Chamados de *erros de transmissão*, os problemas de bits perdidos, modificados ou gerados aleatoriamente são responsáveis por grande parte da complexidade necessária em redes de computadores. Pode-se resumir:

Grande parte da complexidade em redes de computadores surge porque os sistemas de transmissão digitais são suscetíveis à interferência que pode fazer com que dados aleatórios apareçam ou dados transmitidos sejam perdidos ou modificados.

7.7 Bits de paridade e verificação de paridade

Felizmente, poucos sistemas de comunicação sofrem interferência freqüente. Por exemplo, circuitos de comunicação local comumente operam por anos sem problemas sérios. Mais importante, a interferência em circuitos de comunicação de longa distância pode ser tão pequena que os modems têm condições de tratar todos os problemas automaticamente. Apesar da baixa probabilidade de erro, os cientistas e engenheiros que projetam redes entendem que os erros de transmissão realmente acontecem e, por isso, fornecem mecanismos de hardware e software para detectá-los e corrigi-los.

O Capítulo 5 discute um dos mecanismos que o hardware RS-232 usa para detectar erros: quando um caractere começa a chegar, o receptor começa um temporizador e usa o mesmo para verificação do caractere sendo recebido. Se o sinal não permanece em uma voltagem fixa pela duração esperada de cada bit ou se um bit de parada não acontece no tempo apropriado, o hardware declara que aquela interferência causou um erro. Além disso, a maioria dos circuitos RS-232 usa um segundo mecanismo para assegurar que cada caractere chegue intacto. Conhecido como *verificação de paridade (parity checking)*, o mecanismo faz com que o remetente compute um bit adicional, chamado de *bit de paridade (parity bit)*, e anexe-o a cada caractere antes do envio. Após todos os bits de um caractere serem recebidos, o receptor remove o bit de paridade, executa a mesma computação que o remetente e verifica se o resultado está de acordo com o valor do bit de paridade. A computação de paridade é escolhida de forma que, se um dos bits do caractere é danificado em trânsito, a computação do receptor não concordará com o bit de paridade e o receptor indicará que aconteceu um erro.

Existem duas formas de paridade, *par* e *ímpar*; remetente e receptor devem concordar com relação a qual forma está sendo usada. Em uma ou outra, a computação de paridade para um dado caracte-

ter é direta. Para alcançar *paridade par*, o remetente fixa o bit de paridade para 0 ou 1 de forma que faça o número total de bits 1 (inclusive o bit de paridade) um número par. Deste modo, quando estiver usando paridade par, o bit de paridade para 0100101 é 1 porque o caractere contém um número ímpar de bits 1, e o bit de paridade para 0101101 é 0 porque o caractere já contém um número par de bits 1. De forma semelhante, para alcançar *paridade ímpar*, o remetente escolhe um bit de paridade que fará o número total de bits 1 ímpar. Durante a chegada de um caractere, o receptor conta o número de bits 1 para checar a paridade. Se todos os bits do caractere chegam intactos, a computação de paridade do receptor concordará com a do remetente. Se a interferência alterar um dos bits durante a transmissão, a computação do receptor não concordará com o remetente e é o receptor que reportará um erro de paridade.

A paridade ilustra uma idéia importante implementada em vários aspectos de hardware e software de rede:

Para detectar erros, os sistemas de rede normalmente enviam uma quantia pequena de informações adicionais com os dados. Um remetente computa o valor das informações adicionais a partir dos dados, e um receptor executa a mesma computação para se certificar de que o pacote foi transmitido sem erro.

7.8 Probabilidade, matemática e detecção de erros

Embora o mecanismo de paridade discutido funcione ao detectar um único bit errado, ele não pode detectar todos os erros possíveis. Para entender por quê, considere o que acontece se erros de transmissão mudam dois bits de um caractere. Por exemplo, paridade par significa que os bits transmitidos pelo remetente, incluindo o bit de paridade, contém um número par de bits 1. Se erros de transmissão mudam dois bits, existem três casos possíveis: os bits alterados começaram como 0, os bits alterados começaram como 1 ou os bits alterados começaram como um 0 e um 1. Se dois bits 0 são mudados para 1, o número total de bits 1 aumenta em um valor par, então a paridade par é preservada. Analogamente, se dois bits 1 são mudados para 0, a paridade par é preservada porque o número total de bits 1 é diminuído por um número par. Finalmente, se um 1 é mudado para 0 e 0 é mudado para um 1, a paridade é preservada porque o número total de bits 1 permanece o mesmo.

O exemplo mostra que a paridade não pode detectar erros de transmissão que mudem dois bits de um caractere. De fato, a paridade não pode detectar qualquer erro de transmissão que mude um número par de bits. No pior caso, um caractere pode ter todos os bits 1 mudados para 0 e mesmo assim ter paridade par! Pode-se resumir:

Um esquema de paridade, projetado para ajudar a detectar erros de transmissão, envia um bit extra de informação com cada caractere. Embora permita que um receptor determine se um único bit foi mudado, a paridade não pode detectar erros de transmissão que mudam um número par de bits.

Estatísticos e engenheiros analisaram o problema de detecção de erros de transmissão e inventaram vários mecanismos alternativos. Em cada mecanismo, o remetente transmite informações adicionais junto com os dados, e o receptor usa as informações para assegurar que os dados chegaram intactos. As diferenças entre os mecanismos surgem em três formas: o tamanho das informações adicionais (que determina a sobrecarga de transmissão), a complexidade computacional do algoritmo (que determina a sobrecarga computacional exigida para criar ou verificar as informações) e o número de erros de bit que podem ser detectados (que determina quão bem o método pode detectar erros de transmissão). Naturalmente, se o sistema de transmissão subjacente embaralha aleatoriamente os bits, nenhuma quantia de informações adicionais pode absolutamente garantir que os dados chegarão sem erro, pois as informações adicionais podem ser mudadas

tal como os dados originais. Deste modo, todos os métodos de detecção de erro são aproximados – a meta é utilizar um esforço que seja razoável para atingir uma baixa probabilidade de aceitar dados corrompidos.

7.9 Detectando erros com checksums

Embora vários métodos de detecção de erro tenham sido inventados, apenas alguns poucos são usados. Muitos sistemas de rede enviam um *checksum* junto com cada pacote para ajudar o receptor a detectar erros. Para calcular um checksum, o remetente trata os dados como uma seqüência de inteiros binários e computa sua soma. Os dados não são restritos a valores inteiros – podem conter caracteres, números em ponto flutuante ou uma imagem. O sistema de rede trata os dados meramente como uma seqüência de inteiros com o propósito de calcular um checksum. Por exemplo, a Figura 7.6 ilustra o cálculo de um checksum de 16 bits para uma pequena string de texto. Para calcular um checksum, o remetente trata cada par de caracteres como um inteiro de 16 bits e faz a soma. Se a soma cresce mais do que 16 bits, os *bits de transporte (carry bits)* são adicionados à soma final.

Os checksums têm vantagens e desvantagens. As vantagens principais referem-se ao tamanho e à facilidade de computação. A maioria das redes que empregam essa técnica usa um checksum de 16 ou 32 bits e gera um checksum único para um pacote inteiro. O tamanho pequeno do checksum significa que o seu custo de transmissão é muito menor do que o de transmitir os dados. Além disso, como o checksum só exige adição, o processamento necessário para criá-lo ou verificá-lo é pequeno.

O checksum tem a desvantagem de não detectar todos os erros comuns. Por exemplo, a tabela na Figura 7.7 mostra que um checksum não é suficiente para detectar um erro de transmissão que inverte um bit em cada um de quatro itens de dados (para estender o exemplo para um pacote inteiro, imagine que os quatro itens modificados acontecem no meio de vários outros). Apesar das mudanças, um receptor declarará que o pacote tem um checksum válido.

H	e	I	I	o	w	o	r	I	d	.	
48	65	6C	6C	6F	20	77	6F	72	6C	64	2E
4865 + 6C6C + 6F20 + 776F + 726C + 642E + carry = 71FC											

Figura 7.6 Um exemplo de cálculo de checksum de 16 bits para uma string de 12 caracteres ASCII. Os caracteres são agrupados em quantidades de 16 bits, somadas juntas usando aritmética de 16 bits, e os bits de transporte são adicionados ao resultado.

Item de dados em binário	Valor do checksum	Item de dados em binário	Valor do checksum
0001	1	0011	3
0010	2	0000	0
0011	3	0001	1
0001	1	0011	3
totais	7		7

Figura 7.7 Ilustração de como um checksum pode falhar na detecção de erros de transmissão. Inverter o valor do segundo bit em cada item de dados produz o mesmo checksum.

7.10 Detectando erros com verificação de redundância cíclica

Como um sistema de rede pode descobrir mais erros sem aumentar a quantidade de informações adicionais em cada pacote? A resposta reside em técnicas de *Verificação de Redundância Cíclica* (*Cyclic Redundancy Checks, CRC*)⁴, que podem descobrir mais erros do que um checksum. Embora essas técnicas possam ser analisadas matematicamente, a sua simplicidade e elegância podem apenas ser apreciadas entendendo o hardware utilizado para implementá-las. Examinaremos brevemente os componentes básicos de hardware e mostraremos como eles podem ser combinados para produzir um sistema em funcionamento.

O hardware que calcula uma CRC usa dois componentes simples: um *registrator de deslocamento* (*shift register*) e uma unidade *ou exclusivo* (*xor*). A Figura 7.8 mostra o diagrama que se usa para denotar um hardware que produza uma saída igual ao *ou exclusivo* de duas entradas.

O segundo dispositivo de hardware usado para computar uma CRC é uma *registrator de deslocamento*. Pode-se encarar um registrator de deslocamento como um túnel através do qual bits passam em fila única, da direita para a esquerda. O registrator de deslocamento contém um número fixo de bits (por exemplo, um registrator de deslocamento poderia conter 16 bits), de forma que um bit deve sair do registro cada vez que um novo bit entrar. Assume-se também que cada registrator de deslocamento tem uma saída que dá o valor do bit mais à esquerda. Sempre que o bit muda, a saída muda.

Em princípio, um registrator de deslocamento tem duas operações: *initialize* e *desloque*. Quando instruído para inicializar, um registrator de deslocamento deixa todos os bits em zero. Como resultado, sua saída também se torna zero. Quando instruído para deslocar, um registrator de deslocamento move instantaneamente todos os bits à esquerda uma posição, fixa o bit mais à direita de acordo com a entrada atual e configura a saída de acordo com o bit mais à esquerda. A Figura 7.9 ilustra o movimento de valores de bits durante uma operação de deslocamento e como a saída muda.

É importante entender que a mudança do valor de entrada de um registrator de deslocamento não faz com que um valor seja inserido – o registrator de deslocamento só lê sua entrada no momento em que uma operação de deslocamento acontece. Por exemplo, a Figura 6.9a mostra que, embora a entrada seja 1, o bit mais à direita do registrator de deslocamento permanece 0 até que o deslocamento aconteça.

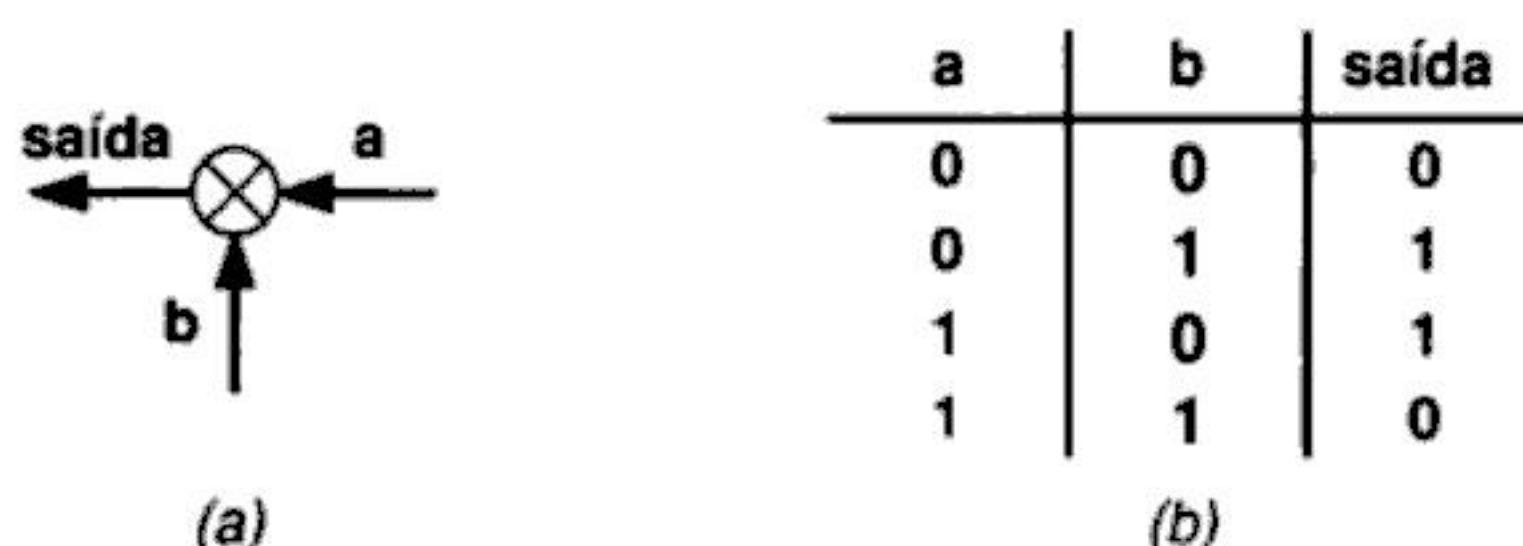


Figura 7.8 (a) Um diagrama de hardware que calcula um *ou exclusivo* e (b) o valor de saída para cada uma das quatro combinações de valores de entrada. Tais unidades de hardware são usadas para calcular uma CRC.

⁴ Além de usadas em redes de computadores, as técnicas CRC também são utilizadas para verificar se os dados foram gravados corretamente em mecanismos de armazenagem, como discos magnéticos.

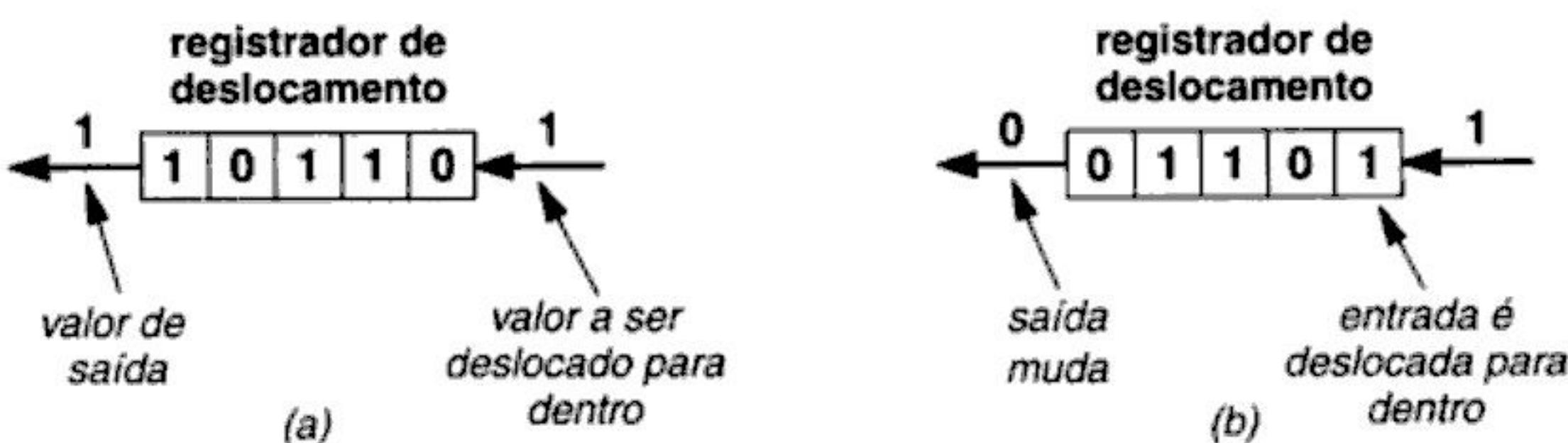


Figura 7.9 Um registrador de deslocamento (a) antes de e (b) depois de uma operação de deslocamento. Durante um deslocamento, cada bit é movido uma posição para a esquerda e a saída se torna igual para o bit mais à esquerda.

7.11 Combinando blocos básicos

A Figura 7.10 mostra como três registradores de deslocamento e três unidades de ou exclusivo podem ser combinados para calcular uma CRC de 16 bits. O hardware é barato e fácil de construir.

Como mostra a figura, o hardware consiste em três registradores de deslocamento interconectados por unidades ou *exclusivos*. A saída da unidade de hardware de ou exclusivo mais à esquerda vai para três lugares simultaneamente: as três unidades de ou exclusivo. Neste exemplo, os registradores de deslocamento contêm 5, 7 e 4 bits. Para calcular uma CRC, os valores em todos os registradores de deslocamento são inicializados com zero e os bits de uma mensagem são deslocados para dentro uma posição de cada vez⁵. Isto é, um bit da mensagem é aplicado à entrada da unidade de ou exclusivo mais à direita, no ponto identificado por *entrada*, e todos os três registradores de deslocamento são instruídos para executar uma operação de deslocamento simultaneamente. O hardware repete o procedimento para cada bit da mensagem. Depois de uma mensagem inteira ter sido deslocada para dentro da unidade, os registradores de deslocamento contêm a CRC de 16 bits para a mensagem. Um receptor usa hardware idêntico para calcular a CRC de uma mensagem sendo recebida e verificar se ele concorda com a CRC que o remetente transmitiu.

Para simplificar a verificação de uma CRC, os algoritmos CRC padrão usam uma pequena modificação em relação ao esquema descrito acima – ao computar uma CRC, o remetente anexa temporariamente 16 bits 0 adicionais à mensagem. Matematicamente, os zeros adicionais fazem com que a CRC resultante aja como um inverso, uma propriedade útil para o receptor. Em vez de computar uma CRC sobre a mensagem recebida e então comparar isso com a CRC recebida, o receptor computa uma CRC sobre a mensagem recebida incluindo a CRC recebida. Se todos os bits forem correta-

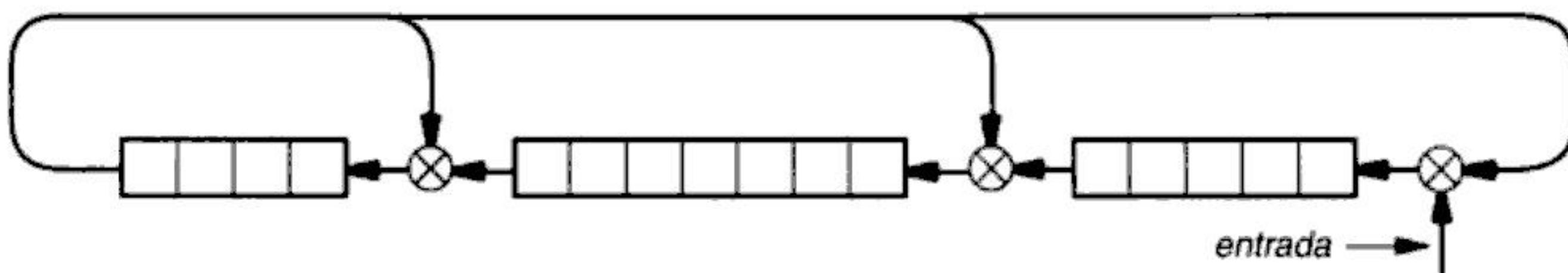


Figura 7.10 Um diagrama do hardware usado para calcular uma CRC. Depois dos bits de uma mensagem terem sido deslocados para a unidade, os registradores de deslocamento contêm a CRC de 16 bits para a mensagem.

⁵ Existem vários algoritmos CRC; eles se diferenciam no número de bits alocados para cada um dos três registradores de deslocamento e os valores iniciais utilizados.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

Exercícios

- 7.1 Assuma que dois computadores estão usando multiplexação por divisão de tempo para alternar o envio de pacotes de 1.000 bytes através de um canal compartilhado que opera a 64.000 bits por segundo. Se o hardware leva 100 microssegundos após um computador parar e antes de outro começar a enviar, quanto tempo levará para que cada um dos computadores envie um arquivo de dados de 1 Megabyte?
- 7.2 No exercício anterior, calcule o tempo exigido para transmissão se os computadores transmitem serialmente. (Assuma um atraso mínimo de 5 microssegundos entre dois pacotes enviados do mesmo computador.)
- 7.3 Considere byte stuffing tal como descrito na Seção 7.5. Encontre uma fórmula que forneça um limite superior no tamanho de dados transferidos como uma função do tamanho dos dados originais.
- 7.4 A técnica de byte stuffing descrita neste capítulo se aplica somente à multiplexação? Para descobrir, crie um multiplexador e demultiplexador que use byte stuffing. Faça com que o transmissor aceite fluxos de dados de 8 bits de três fontes, envie os dados através de uma interconexão compartilhada e então separe-os em três fluxos independentes.
- 7.5 Um sistema de byte stuffing é *adaptativo* se o remetente pode escolher o caractere de escape para cada transferência. Para informar o receptor sobre o esquema que está sendo usado, o remetente transmite uma sequência como *soh x*, onde *x* denota o caractere de escape a ser usado. O valor de *x* é escolhido para minimizar o tamanho da transferência. Projete um algoritmo que toma como entrada um arquivo de dados para ser transmitido e produza um esquema de byte stuffing que minimiza os dados transferidos.
- 7.6 Escreva um programa de computador que faça byte stuffing adaptativo como descrito no exercício anterior. Execute seu programa em arquivos binários. Como o tamanho dos dados transferidos se compara aos dados transferidos com um esquema não-adaptável? Você alguma vez envia mais dados?
- 7.7 Suponha que um hardware defeituoso em um sistema de transmissão orientado a caractere configure todos os bits transferidos em zero. Um bit de paridade identifica o problema? Por que ou por que não?
- 7.8 Escreva um programa de computador que compute um checksum de 16 bits sobre um arquivo. O método mais rápido usa um inteiro de 32 bits para acumular a soma. Lembre-se de tratar os bits de vai-um (i.e., bits além dos primeiros 16) adicionando-os de volta na soma.
- 7.9 No exercício anterior, o que acontece se o fato de somar bits de vai-um causar outro vai-um? Como a situação pode ser eficazmente resolvida?
- 7.10 Escreva um programa de computador para simular o hardware CRC mostrado na Figura 7.10.
- 7.11 O cálculo de checksum é freqüentemente implementado com software convencional de computador, mas a maioria das computações de CRC são executadas com hardware dedicado. Por quê?

Sumário do Capítulo

- 8.1** Introdução 115
- 8.2** Comunicação Direta Ponto a Ponto 115
- 8.3** Canais de Comunicação Compartilhados 117
- 8.4** Importância das LANs e Localidade de Referência 118
- 8.5** Topologias de LAN 118
- 8.6** Exemplo de Rede de Barramento: Ethernet 120
- 8.7** Detecção de Portadora em Redes de Acesso Múltiplo (Carrier Sense on Multi-Access Networks, CSMA) 122
- 8.8** Detecção de Colisões e Backoff com CSMA/CD 123
- 8.9** LANs Sem Fio 802.11b e CSMA/CA 124
- 8.10** Outro Exemplo de Rede de Barramento 125
- 8.11** Topologia em Anel e Passagem de Token 125
- 8.12** Redes de Passagem de Token de Autocura 127
- 8.13** Exemplo de Rede Estrela: ATM 128
- 8.14** Resumo 129

Tecnologias de LAN e Topologias de Rede

8.1 Introdução

Os capítulos anteriores explicam como os bits podem ser codificados em sinais que são transmitidos através de um canal de comunicação e discutem o hardware de modem que interconecta dois computadores e executa a codificação e decodificação necessária. Embora sejam necessários modems para a comunicação de longa distância, a maioria das redes são locais – a rede cabe em um edifício ou uma sala. Além disso, redes pequenas são projetadas freqüentemente para permitir que múltiplos computadores compartilhem recursos. Por exemplo, uma rede local que conecte dois computadores e uma impressora permite a qualquer um dos computadores acessar a impressora.

As tecnologias de hardware usadas para redes locais não consistem em modems separados e cabos. Em vez disso, as tecnologias são projetadas visando ao compartilhamento. Elas permitem que múltiplos computadores e dispositivos, como impressoras, sejam conectados diretamente a uma rede única e compartilhada. Como o meio subjacente é compartilhado, os computadores devem se alternar no uso do mesmo. Este capítulo descreve os conceitos de redes locais subjacentes e explica por que as redes que usam compartilhamento se tornaram populares. Descreve também topologias de rede básicas e examina exemplos de tecnologias de redes locais populares e históricas¹. Os próximos três capítulos continuam a discussão de redes usadas para comunicação local descrevendo detalhes adicionais, incluindo uma explicação de esquemas de cabeamento.

8.2 Comunicação direta ponto a ponto

Todos os sistemas primitivos de comunicação entre computadores usaram o padrão descrito nos capítulos anteriores. Cada canal de comunicação (por exemplo, um circuito de dados alugado) conectava exatamente dois computadores e estava disponível exclusivamente para aqueles computadores. Conhecido como *rede ponto a ponto* ou *rede de malha (mesh network)*, o esquema tem três propriedades úteis. Primeiro, pode ser usado hardware apropriado, pois cada conexão está inde-

¹ Embora atualmente uma tecnologia de LAN em particular domine o mercado comercial, tecnologias mais antigas de LAN ilustram melhor alguns conceitos e alternativas.

pendentemente instalada. Por exemplo, a capacidade de transmissão, ou seja, a largura de banda, do circuito subjacente e os modems usados não precisam ser os mesmos em todas as conexões. Segundo, como têm acesso exclusivo, os computadores conectados podem decidir exatamente como enviar dados através da conexão. Eles podem escolher um formato de quadro, um mecanismo de detecção de erro e um tamanho máximo de quadro. Mais importante, já que cada conexão é independente das demais, os detalhes podem ser mudados sempre que os proprietários dos computadores concordarem em realizar uma mudança. Terceiro, como apenas dois computadores têm acesso ao canal, é fácil reforçar a segurança e a privacidade. Nenhum outro computador manipula dados, e, portanto, nenhum outro computador pode obter acesso.

Evidentemente, as conexões ponto a ponto têm desvantagens também. A principal se torna aparente quando mais de dois computadores precisam se comunicar um com o outro. Em um esquema ponto a ponto, que fornece um canal de comunicação separado para cada par de computadores, o número de conexões cresce depressa ao aumentar o conjunto. Por exemplo, a Figura 8.1 mostra que dois computadores precisam de apenas uma conexão, três computadores precisam de três conexões e quatro computadores precisam de seis conexões.

Como a figura ilustra, o número total de conexões cresce mais rapidamente que o número total de computadores. Matematicamente, o número de conexões necessárias para N computadores é proporcional à raiz de N :

$$\text{conexões diretas necessárias} = \frac{(N^2 - N)}{2}$$

Intuitivamente, pode-se entender o efeito considerando o quanto caro se torna adicionar um novo computador a um conjunto existente: um novo computador adicionado deve ter uma conexão para cada um dos computadores já existentes. Deste modo, a soma do N -ésimo computador exige $N - 1$ conexões novas.

Na prática, o custo é especialmente alto porque muitas conexões seguem o mesmo caminho físico. Por exemplo, suponha que uma organização tenha cinco computadores, estando dois em uma localização (por exemplo, o andar térreo de um edifício) e três em outra (por exemplo, o último andar do mesmo edifício). A Figura 8.2 mostra que, se cada computador tem uma conexão para todos outros computadores, seis conexões passam entre as duas localizações – em muitos casos, tais conexões seguem o mesmo caminho físico.

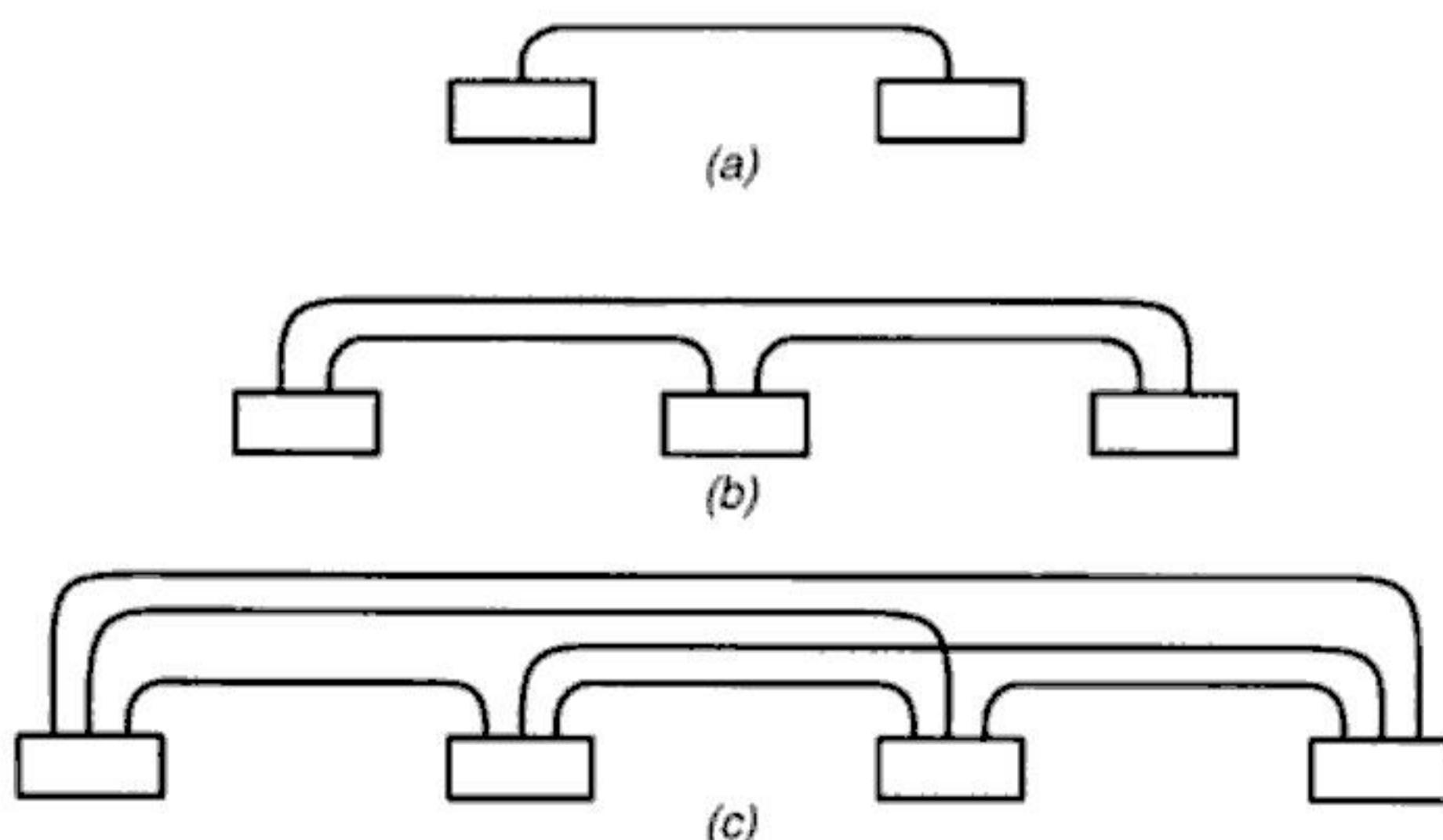


Figura 8.1 As conexões independentes ponto a ponto necessárias para (a) dois, (b) três e (c) quatro computadores. O número de conexões cresce rapidamente à medida que aumenta o número de computadores.

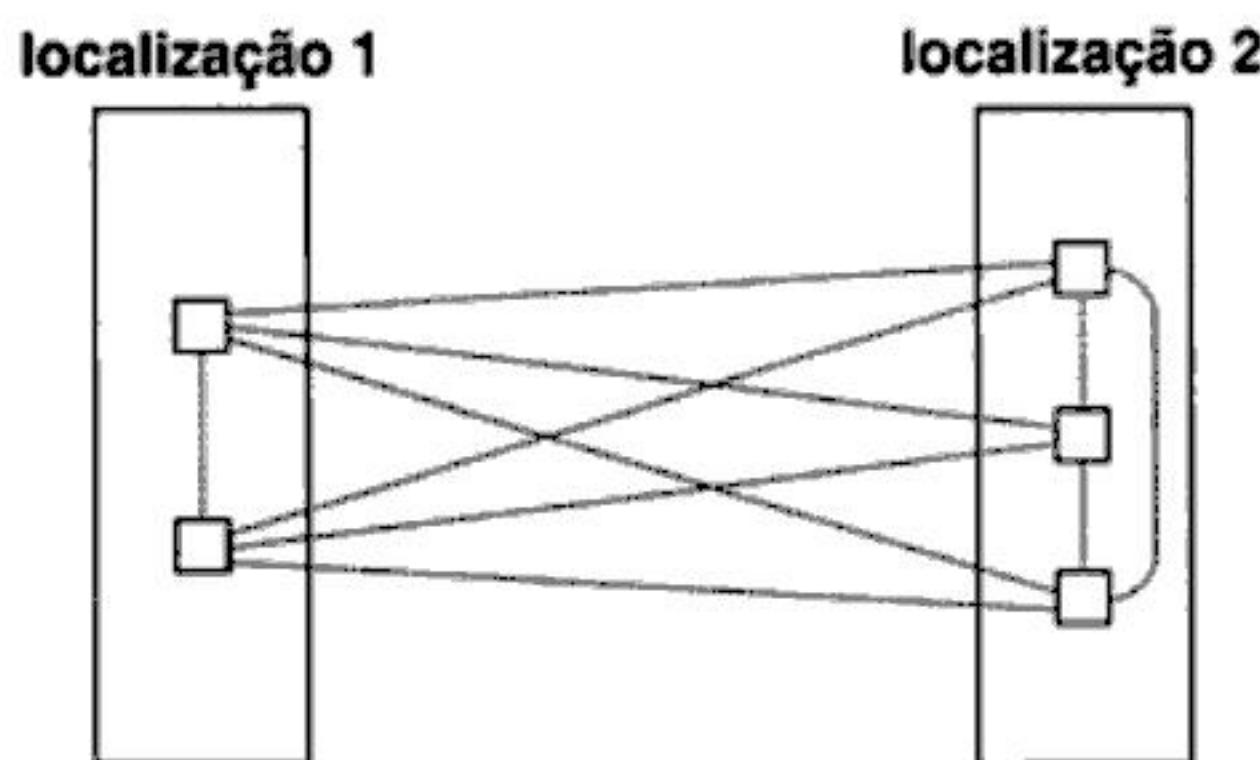


Figura 8.2 A desvantagem de uma rede ponto a ponto que exige uma conexão dedicada para cada par de computadores: o número total de conexões que passa entre duas localizações pode exceder o número total de computadores que estão sendo conectados.

A figura mostra que, em uma rede ponto a ponto, o número de conexões passando entre duas localizações normalmente excede o número total de computadores. Se um outro computador é adicionado aos dois computadores na localização 1, a situação piora: o número total de computadores se torna seis e o número de conexões que passa entre as duas localizações aumenta para nove².

8.3 Canais de comunicação compartilhados

A história da ligação de computadores em rede mudou drasticamente durante o final dos anos 60 e início dos 70 quando alguns investigadores desenvolveram formas de comunicação entre computadores conhecida como *Redes Locais (Local Area Networks, LANs)*. Inventado como alternativa para as caras conexões dedicadas ponto a ponto, o projeto difere fundamentalmente de redes de longa distância porque se baseia no compartilhamento da rede. Cada LAN consiste em um meio compartilhado único, normalmente um cabo, em que muitos computadores se acoplam. Os computadores se alternam no uso do meio para enviar pacotes.

Vários projetos de LAN emergiram da pesquisa. Os projetos diferem em detalhes como as voltagens e técnicas de modulação usadas e a abordagem de compartilhamento (ou seja, os mecanismos usados para coordenar o acesso e transmitir pacotes).

Uma vez que o compartilhamento elimina duplicação, ele tem um impacto econômico importante em redes: reduz o custo. Consequentemente, as tecnologias de Rede Local que permitem que um conjunto de computadores compartilhe um meio se tornou popular. De fato,

As redes que permitem que múltiplos computadores compartilhem um meio de comunicação são usadas para comunicação local. As conexões ponto a ponto são usadas para redes de longa distância e para alguns outros casos especiais.

Se o compartilhamento reduz o custo, por que redes compartilhadas são usadas somente para comunicação local? Tanto razões técnicas quanto econômicas contribuem para isso. Foi dito que os computadores acoplados a uma rede compartilhada devem coordenar o uso da rede. Como a coordenação exige comunicação e o tempo para se comunicar depende da distância, uma separação geográfica grande entre computadores introduz atrasos mais longos. Deste modo, as redes compartilhadas com longos atrasos são ineficientes porque gastam mais tempo coordenando o uso do meio compartilhado e menos tempo enviando dados. Além disso, os engenheiros descobriram que

² O Capítulo 13 descreve como pode ser usada a comutação eletrônica para reduzir o número de conexões.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

8.5.3 Topologia de barramento

Uma rede que usa uma *topologia de barramento* normalmente consiste em um único cabo longo ao qual computadores se acoplam⁴. Qualquer computador acoplado a um barramento pode enviar um sinal através do cabo, e todos os computadores receberão esse sinal. A Figura 8.5 ilustra a topologia. Como todos os computadores ligados pelo cabo podem detectar um sinal elétrico, qualquer computador pode enviar dados a qualquer outro computador. Naturalmente, os computadores acoplados a uma rede de barramento devem se coordenar para assegurar que somente um computador envie um sinal a cada momento, evitando o caos.

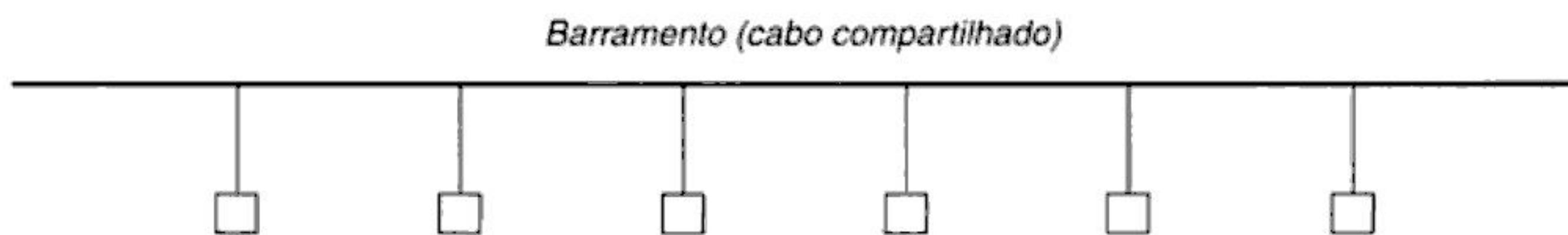


Figura 8.5 Ilustração de uma topologia de barramento em que todos os computadores estão ligados a um cabo único.

8.5.4 A razão para topologias múltiplas

Cada topologia tem vantagens e desvantagens. Uma topologia em anel torna mais fácil aos computadores coordenarem o acesso e detectarem se a rede está operando corretamente. Porém, uma rede inteira em anel é desativada se um dos cabos é cortado. Uma topologia em estrela ajuda a proteger a rede de danos em um único cabo, já que cada cabo conecta somente uma máquina. Um barramento exige menos fios que uma estrela, mas tem a mesma desvantagem de um anel: uma rede é desativada se alguém accidentalmente corta o cabo principal. Além de seções posteriores neste capítulo, outros capítulos fornecem exemplos detalhados de tecnologias de rede que mostram algumas das diferenças.

Pode-se resumir os pontos mais importantes sobre topologias de rede como:

As redes são classificadas em categorias abrangentes de acordo com sua forma geral. As topologias primárias usadas com LANs são estrela, anel e barramento; cada topologia tem vantagens e desvantagens.

8.6 Exemplo de rede de barramento: Ethernet

8.6.1 História da Ethernet

Ethernet é uma tecnologia de rede bem-conhecida e extensamente usada que emprega topologia de barramento. A Ethernet foi inventada no Centro de Pesquisa da Corporação Xerox em Palo Alto, no início dos anos 70. Mais tarde, Digital Equipment Corporation, Intel Corporation e Xerox cooperaram para desenvolver um padrão de produção, informalmente chamado de *DIX Ethernet*, devido às iniciais das três empresas. O IEEE agora controla os padrões Ethernet⁵. Em sua versão original, uma LAN Ethernet consistia em um cabo coaxial único, chamado *éter*, ao qual múltiplos computa-

⁵ Existem diversas variações de Ethernet; o Capítulo 10 discute as alternativas.

⁴ Na prática, as extremidades da rede de barramento devem ser terminadas para evitar que um sinal elétrico seja refletido de volta ao barramento.

dores se conectam. Os engenheiros usam o termo *segmento* para se referir ao cabo coaxial de Ethernet. Um dado segmento de Ethernet é limitado a 500 metros de comprimento, e o padrão exige uma separação mínima de 3 metros entre cada par de conexões.

O hardware Ethernet original operava em uma largura de banda de 10 Megabits por segundo (Mbps); uma versão posterior conhecida como *Fast Ethernet* opera a 100 Mbps, e a versão mais recente, conhecida como *Gigabit Ethernet*, opera a 1.000 Mbps ou 1 Gigabit por segundo (Gbps).

8.6.2 Transmissão pela Ethernet e Codificação Manchester

O padrão Ethernet especifica todos os detalhes, inclusive o formato de quadros que os computadores enviam através do éter⁶, a voltagem a ser usada e o método usado para modular um sinal. Por exemplo, o padrão especifica que quadros são enviados utilizando a *Codificação Manchester*.

Para entender a Codificação Manchester é necessário saber que o hardware pode notar uma mudança na voltagem mais facilmente do que um valor fixo. Como resultado, em vez de codificar valores usando duas voltagens, como o RS-232, os valores são codificados como uma série de mudanças⁷. Tecnicamente, diz-se que o hardware é *disparado pelo limite* (*edge triggered*), e as mudanças são conhecidas como *limites de subida* (*rising edge*) ou *descida* (*falling edge*). A Codificação Manchester usa subidas e descidas para codificar dados. O transmissor envia uma descida para codificar 0 e uma subida para codificar 1. A Figura 8.6 ilustra a codificação.

Na figura, o eixo x representa o tempo necessário para transmitir os 12 bits 101011000101, e o eixo y representa a voltagem. O eixo x é dividido em 12 (fatias) "slots" correspondentes a um único bit. A mudança de voltagem que codifica um valor digital ocorre exatamente no meio dessas fatias. Por exemplo, quando a transmissão tem início, a voltagem é zero. Exatamente na metade da primeira fatia, a voltagem se torna positiva para codificar o valor de bit 1. De forma similar, exatamente na metade da segunda fatia de tempo, a voltagem desce para codificar um valor de 0. Se dois bits justapostos têm o mesmo valor, uma mudança adicional de voltagem ocorre no final da fatia. Por exemplo, uma vez que uma subida codifica um valor 1, quando dois bits de valor 1 estiverem justapostos a voltagem precisa cair no limite entre esses dois bits.

Para entender o significado do sinal, o receptor precisa saber exatamente quando inicia e termina cada fatia. A Codificação Manchester usa um *préambulo* para permitir tal sincronização. O préam-

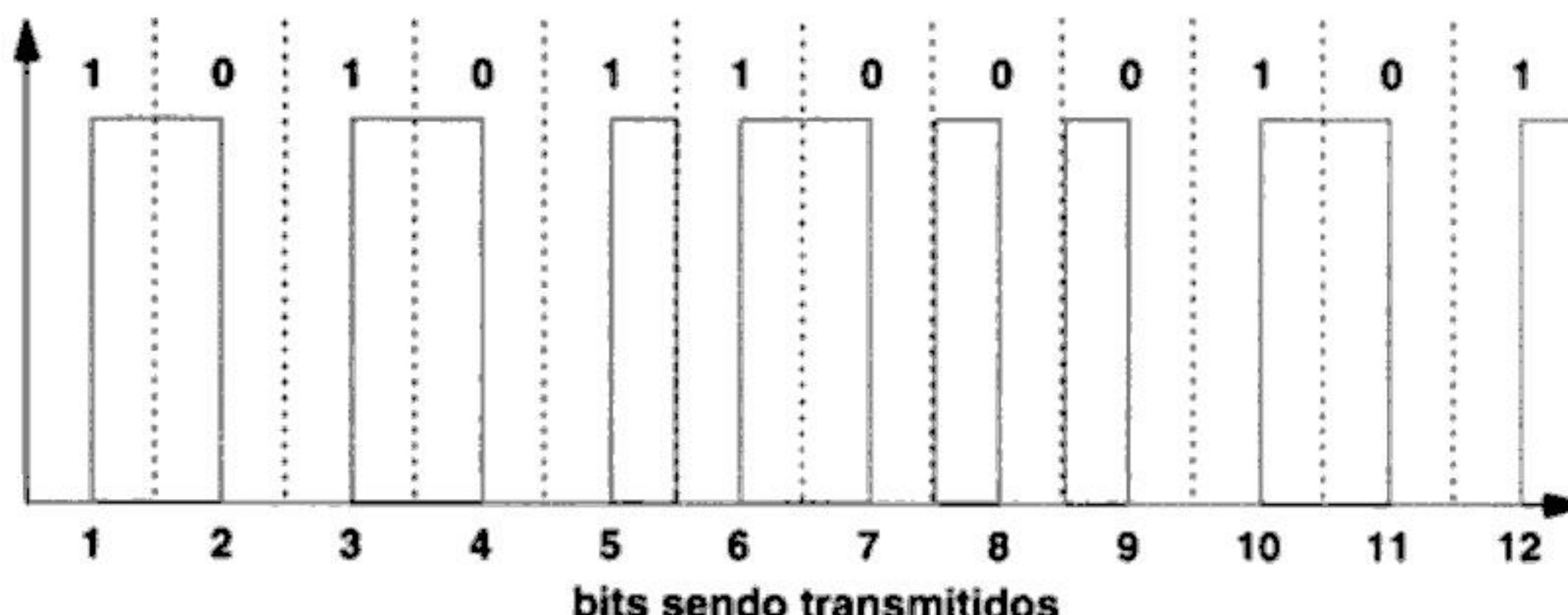


Figura 8.6 Ilustração da Codificação Manchester utilizada com a Ethernet. Uma mudança de voltagem positiva para voltagem zero codifica um bit de valor 0, e uma mudança de zero para positivo codifica um bit de valor 1.

⁶ O Capítulo 9 discute quadros Ethernet em mais detalhes e fornece um exemplo.

⁷ A Figura 5.2 na página 60 ilustra a codificação do RS-232.

bulo consiste em 64 zeros e uns alternados, enviados antes do quadro. Como a figura ilustra, alternar zero e um produz uma onda de margens quadradas com a transmissão exatamente no meio de cada fatia. Hardwares de recepção usam o preâmbulo para sincronizar sua noção das fatias de tempo com o sinal recebido. Uma vez que o preâmbulo tenha sido processado, o receptor pode identificar corretamente bits sucesivos.

8.6.3 Compartilhamento em uma Ethernet

Como a Ethernet usa uma topologia de barramento, múltiplos computadores devem compartilhar o acesso a um meio único. Um remetente transmite um sinal, que se propaga em direção às duas extremidades do cabo. A Figura 8.7 mostra como os dados fluem através de uma Ethernet.

Como mostra a figura, um sinal se propaga do computador remetente até as duas extremidades do cabo compartilhado. É importante entender que compartilhamento em tecnologias de redes locais não significa que estão sendo enviados múltiplos quadros ao mesmo tempo. Em vez disso, o computador remetente tem uso exclusivo do cabo inteiro durante a transmissão de um dado quadro – os outros computadores devem esperar. Depois de um computador terminar de transmitir um quadro, o cabo compartilhado se torna disponível para outro computador usar. Para resumir:

Ethernet é uma rede de barramento em que múltiplos computadores compartilham um meio de transmissão único. Enquanto um computador transmite um quadro para outro, todos os demais devem esperar.

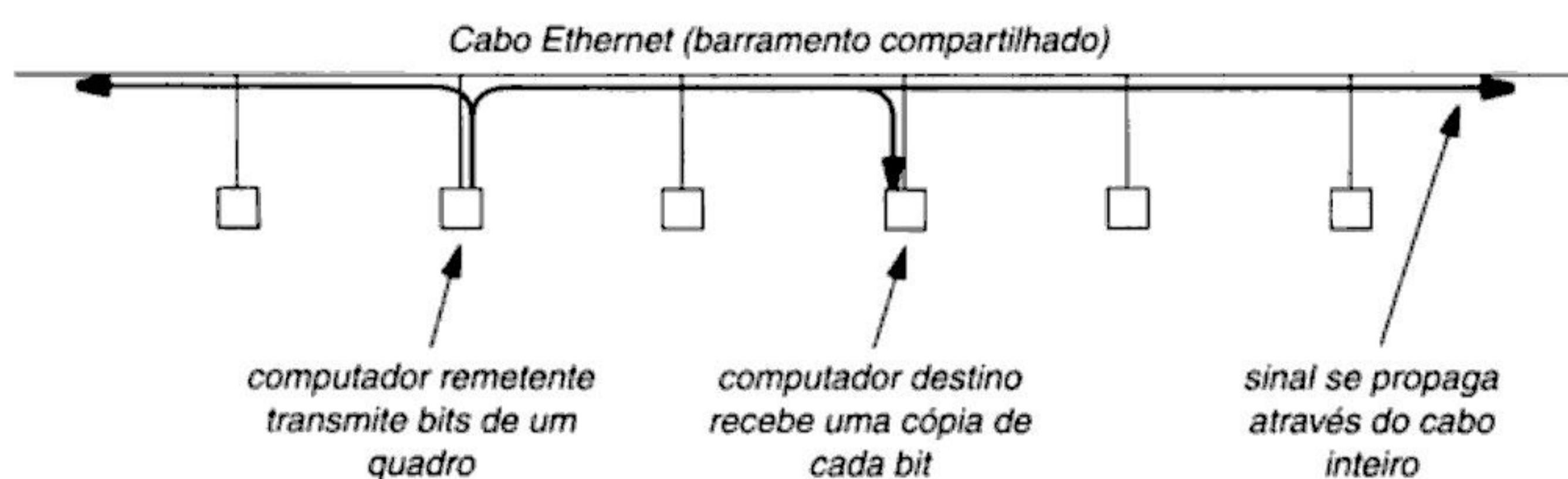


Figura 8.7 Fluxo conceitual de bits através de uma Ethernet. Enquanto estiver transmitindo um quadro, o computador tem uso exclusivo do cabo.

8.7 Detecção de portadora em redes de acesso múltiplo (*Carrier Sense Multiple Access, CSMA*)

O aspecto mais interessante sobre a Ethernet é o mecanismo usado para coordenar a transmissão. Uma rede Ethernet não tem uma controladora centralizada que diz a cada computador como se alternar usando o cabo compartilhado. Todos os computadores acoplados a uma Ethernet participam em um esquema de coordenação distribuída, chamado *Carrier Sense Multiple Access (CSMA)*. O esquema usa atividade elétrica no cabo para determinar o estado. Quando nenhum computador estiver enviando um quadro, o éter não contém sinais elétricos. Durante a transmissão de um quadro, porém, um remetente transmite sinais elétricos usados para codificar bits. Embora os sinais sejam ligeiramente diferentes das ondas de portadora descritas no Capítulo 5, eles são informalmente chamados de *portadora*. Deste modo, para determinar se o cabo está em uso, um computador pode verificar se há uma portadora. Se não houver, pode transmitir um quadro. Se houver, ele

deve esperar que o remetente termine antes de continuar. Tecnicamente, procurar por uma onda de portadora é chamado de *sensor da portadora (carrier sense)*, e a idéia de usar a presença de um sinal para determinar quando transmitir é chamada de *Carrier Sense Multiple Access (CSMA)*.

8.8 Detecção de colisões e backoff com CSMA/CD

Como o CSMA permite que cada computador determine se um cabo compartilhado já está em uso por outro computador, ele previne que um computador interrompa uma transmissão em andamento. Porém, o CSMA não pode prevenir todos os conflitos possíveis. Para entender por quê, imagine o que acontece se dois computadores em extremidades opostas de um cabo inativo têm um quadro pronto para enviar ao mesmo tempo. Quando eles tentam detectar uma portadora, ambas as estações encontram o cabo inativo e começam a enviar quadros simultaneamente. Os sinais viajam a aproximadamente 70% da velocidade da luz, e quando os sinais transmitidos por dois computadores alcançam o mesmo ponto no cabo, eles interferem um no outro.

A interferência entre dois sinais é chamada de *colisão*. Embora uma colisão não prejudique o hardware, ela produz uma transmissão adulterada que previne que qualquer um dos quadros seja corretamente recebido. Para assegurar que nenhum outro computador transmita simultaneamente, o padrão Ethernet exige que estações remetentes monitorem sinais no cabo. Se o sinal no cabo difere do sinal que a estação está enviando, quer dizer que uma colisão ocorreu⁸. Sempre que uma colisão é detectada, a estação remetente pára imediatamente de transmitir. Tecnicamente, a monitoração de um cabo durante a transmissão é conhecida como *Detecção de Colisão (Collision Detection, CD)* e o mecanismo para Ethernet é conhecido como *Carrier Sense Multiple Access with Collision Detection (CSMA/CD)*.

O CSMA/CD faz mais do que apenas detectar colisões – ele também se recupera delas. Depois de acontecer uma colisão, um computador deve esperar que o cabo se torne novamente inativo antes de transmitir um quadro. Porém, se os computadores começarem a transmitir assim que o éter se tornar inativo, outra colisão acontecerá. Para evitar colisões múltiplas, a Ethernet exige que cada computador atrasse após uma colisão e antes de tentar uma retransmissão. O padrão especifica um atraso máximo, d , e força cada computador a escolher um atraso aleatório menor que d . Na maioria dos casos, quando um computador escolhe um atraso aleatoriamente, ele selecionará um valor que diferirá de quaisquer outros valores escolhidos pelos demais computadores – o computador que escolhe o menor atraso prosseguirá com o envio de um quadro e a rede retornará à operação normal.

Se acontecer de dois ou mais computadores escolherem quase a mesma quantia de atraso após uma colisão, eles começarão a transmitir quase ao mesmo tempo, produzindo uma segunda colisão. Para evitar uma seqüência de colisões, a Ethernet exige que cada computador dobre a faixa de valores a partir da qual um atraso é escolhido após cada colisão. Deste modo, um computador escolhe um atraso aleatório de 0 até d após uma colisão, um atraso aleatório entre 0 e $2d$ após uma segunda colisão, entre 0 e $4d$ após uma terceira e assim por diante. Depois de algumas colisões, a faixa em que um valor aleatório é escolhido se torna grande, e a probabilidade de que algum computador escolha um atraso pequeno e transmita sem colisão é alta.

Tecnicamente, dobrar a faixa do atraso aleatório após cada colisão é conhecido como *backoff exponencial binário*. Em essência, backoff exponencial significa que uma Ethernet pode se recuperar depressa após uma colisão porque cada computador concorda em esperar por tempos maiores entre tentativas quando o cabo ficar ocupado. No caso improvável de dois ou mais computadores escolherem atrasos aproximadamente iguais, o backoff exponencial garante que a disputa pelo cabo será reduzida após algumas colisões. Pode-se resumir:

⁸ Para garantir que uma colisão tenha tempo de alcançar todas as estações antes de parar de transmitir, o padrão Ethernet especifica tanto um comprimento de cabo máximo quanto um tamanho de quadro mínimo.

Computadores ligados a uma Ethernet usam CSMA/CD na qual um computador espera pela inatividade do éter antes de transmitir um quadro. Se dois computadores transmitem simultaneamente, acontece uma colisão; os computadores usam backoff exponencial para escolher qual computador prosseguirá. Cada computador demora um tempo aleatório antes de tentar transmitir novamente e então dobra a demora para cada colisão sucessiva.

8.9 LANs sem fio 802.11b e CSMA/CA

Existe um conjunto de tecnologias de *LAN sem fio* que usam uma forma modificada de CSMA/CD. Os produtos, que são fabricados por várias empresas, estão disponíveis sob uma variedade de nomes comerciais. Por exemplo, a Apple Computer Corporation vende um dispositivo de *Airport*; a Lucent Corporation vende *WaveLan*; a Solectek vende *AirLAN*; e a Proxim Corporation vende *RangeLAN*. Dispositivos mais antigos usam freqüências de 900MHz para permitir que os dados sejam enviados a 2Mbps; o padrão 802.11b da IEEE, também conhecido como *Wi-Fi*, define LANs sem-fio que operam a 11Mbps usando uma freqüência com alcance de 2,4GHz. Um padrão conhecido como *bluetooth* especifica uma tecnologia de LAN sem fio designada para curtas distâncias.

Em vez de transmitir sinais através de um cabo, o hardware de LAN sem fio usa antenas para transmitir sinais de RF através do ar, que outros computadores recebem. Como outras tecnologias de LAN, as LANs sem fio usam compartilhamento. Isto é, todos os computadores que participam em uma determinada LAN sem fio são configurados para usar uma mesma freqüência de rádio. Deste modo, eles devem se alternar no envio de pacotes.

Uma diferença entre o modo como LANs com e sem fio administram compartilhamento surge por causa da forma com que as transmissões sem fio se propagam. Embora a energia eletromagnética se irradie em todas as direções, os transmissores de LAN sem fio usam pouca energia, o que significa que uma transmissão tem energia suficiente somente para viajar uma distância pequena. Além disso, obstruções metálicas podem bloquear o sinal. Assim, as unidades sem fio localizadas em pontos bem distantes ou atrás de obstruções não receberão as transmissões do outro.

A falta de comunicação completa significa que as LANs sem fio não podem usar o mesmo mecanismo de CSMA/CD que a Ethernet usa. Para entender por quê, considere três computadores com hardware de LAN sem fio posicionados longe um do outro, como mostra a Figura 8.8.

Na figura, os dois computadores de fora estão muito longe um do outro para receber suas transmissões. Em tais situações, escuta de portadora e detecção de colisões não bastam. Por exemplo, suponha que o computador 1 esteja enviando um pacote para o computador 2. Já que o computador 3 não pode receber a transmissão, ele poderia prosseguir com a transmissão, resultando em uma colisão. De forma semelhante, se os computadores 1 e 3 transmitissem um quadro ao mesmo tempo, somente o computador 2 poderia detectar uma colisão⁹.

Para assegurar que eles compartilham os meios de transmissão corretamente, as LANs sem fio usam um esquema modificado conhecido como *Carrier Sense Multiple Access with Collision Avoidance* (CSMA/CA). Em vez de depender de todos os outros computadores para receber todas as transmissões, o CSMA/CA usado com LANs sem fio ativa uma breve transmissão do receptor pretendido antes de transmitir um pacote. Por exemplo, suponha que o computador 1 na Figura 8.8 precisa enviar um quadro para o computador 2. Antes de enviá-lo, o computador 1 transmite uma breve mensagem de controle. Quando o computador 2 a recebe, responde enviando outra mensagem de controle para indicar que está pronto para receber uma transmissão. Quando o computador 1 recebe a resposta de seu receptor pretendido, ele começa a transmissão do quadro.

⁹ O problema também é conhecido como *problema da estação escondida (hidden station problem)*.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

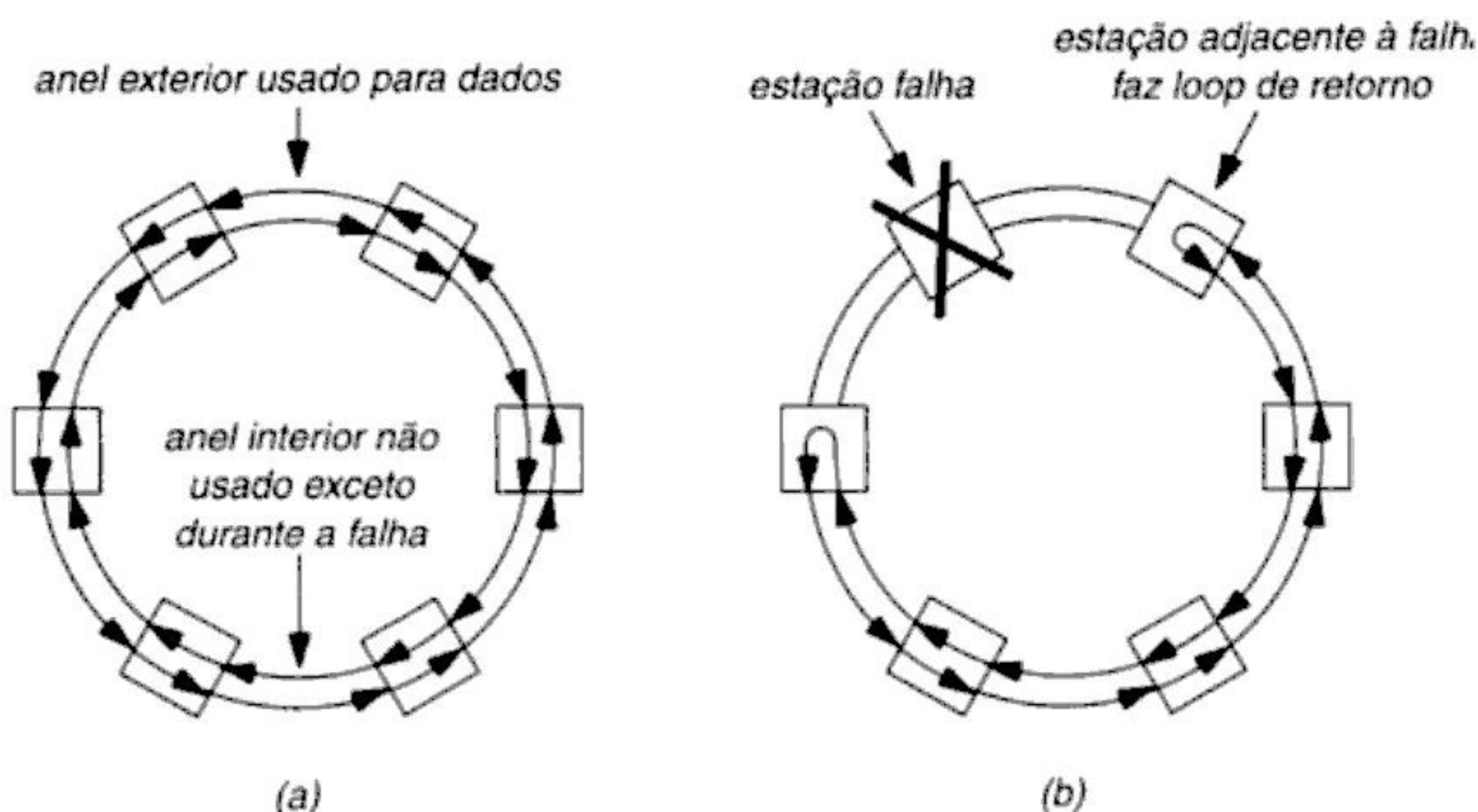


Figura 8.10 (a) Uma rede FDDI com setas mostrando as direções dos fluxos de dados e (b) a mesma rede depois de uma estação falhar. Normalmente, os dados viajam em uma direção. Depois de uma estação falhar, as estações adjacentes usam o caminho contrário para formar um anel fechado.

Uma tecnologia de rede que usa anéis de rotação contrária é chamada de autocura (self healing) porque o hardware pode detectar uma falha catastrófica e se recuperar automaticamente. Um anel é usado para transmitir dados. Quando acontece um defeito que o desliga, as estações adjacentes ao defeito se reconfiguram automaticamente, usando o segundo anel para contornar a falha.

8.13 Exemplo de rede estrela: ATM

As companhias telefônicas desenvolveram uma tecnologia em rede conhecida como *Modo de Transferência Assíncrona (Asynchronous Transfer Mode, ATM)*¹¹. O elemento básico de uma rede ATM é um switch eletrônico ao qual vários computadores podem se conectar. Por exemplo, a Figura 8.11 mostra seis computadores conectados a um switch ATM.

A figura mostra porque o ATM é classificado como uma topologia em estrela. Um ou mais switches interconectados formam um hub central ao qual todos os computadores estão ligados. Diferentemente de topologias de barramento ou anel, uma rede em estrela não propaga dados para outros computadores além dos pares comunicantes – o hub recebe dados diretamente dos remetentes e transmite dados diretamente para o receptor. Note que a topologia em estrela torna uma rede ATM menos dependente das conexões com computadores individuais do que uma rede que usa uma topologia em anel. Se a conexão entre um computador e o switch é perdida, somente aquele computador é afetado.

Uma vez que o ATM é projetado para fornecer alta performance, uma conexão típica entre um computador e um switch ATM opera em uma velocidade de 155 Mbps ou mais. Para transportar taxas de dados tão altas, a conexão entre um computador e um switch ATM normalmente usa fibra óptica em vez de fios de cobre. Na realidade, cada conexão usa um par de fibras como representado na Figura 8.12. Uma única fibra é utilizada para carregar dados em uma direção.

¹¹ O capítulo 14 cobre o ATM com mais detalhes.

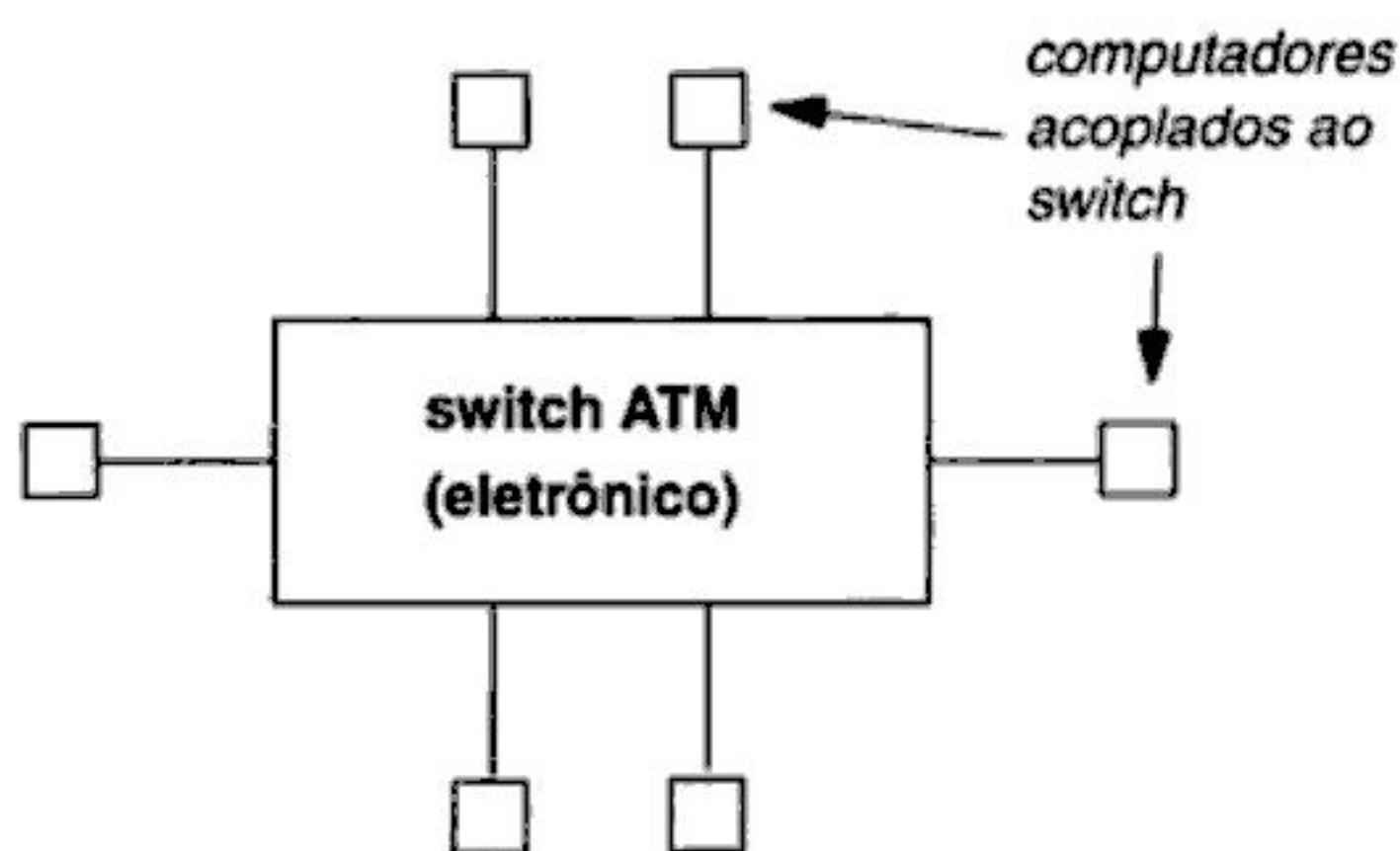


Figura 8.11 Um switch ATM com seis computadores acoplados e a topologia de estrela que resulta dele.

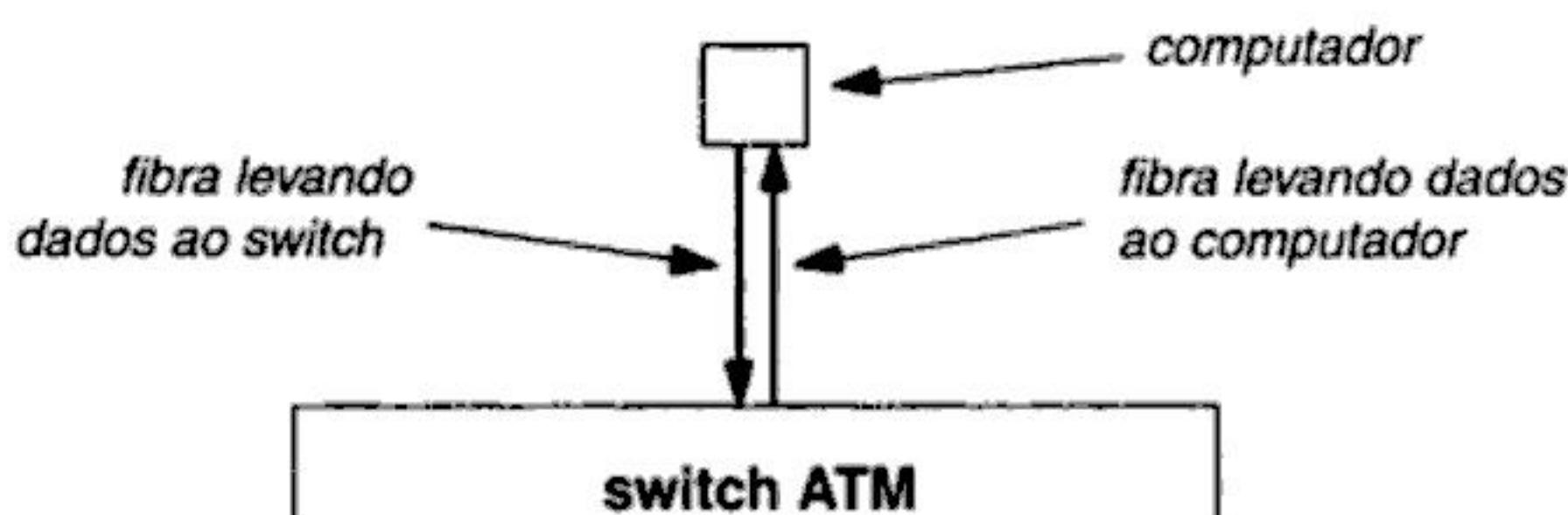


Figura 8.12 Detalhes de uma conexão entre um switch ATM e um computador. Cada conexão consiste em um par de fibras ópticas. Uma fibra carrega dados para o switch e a outra carrega dados para o computador.

As fibras do par usado para conectar um computador a um switch ATM são presas uma à outra. Geralmente, o invólucro de uma fibra tem uma faixa colorida ou é etiquetado; quem quer que instale uma conexão usa a etiqueta para garantir que a saída do switch se conecta à entrada do computador, e vice-versa.

Para resumir:

Uma rede ATM é formada por um switch ao qual se ligam múltiplos computadores. A conexão entre um computador e um switch ATM consiste em um par de fibras, cada uma transportando dados em uma direção.

8.14 Resumo

Este capítulo discute uma alternativa para comunicação direta ponto a ponto chamada Rede Local (LAN). Projetada para uso em uma distância pequena (por exemplo, em um edifício), uma LAN não precisa de um fio separado entre cada par de computadores. Uma LAN consiste em um meio único, compartilhado, ao qual se ligam muitos computadores. Os computadores se alternam no uso do meio para enviar dados.

Embora as tecnologias de LAN exijam que os computadores dividam dados em pequenos pacotes chamados quadros, somente um pacote pode ser transmitido em uma LAN de cada vez. Isto é, quando estiver transmitindo, um computador tem uso exclusivo da LAN. Para obter acesso justo, cada computador tem permissão para segurar o meio compartilhado durante a transmissão de um

quadro antes de permitir que outro computador prossiga. Deste modo, após ganhar controle, um computador envia um quadro e então renuncia o controle para outro computador.

Cada rede de computadores pode ser classificada em alguma das poucas categorias básicas, dependendo de sua topologia. Uma topologia de barramento consiste em um cabo único, compartilhado, ao qual muitos computadores se ligam. Quando usa um barramento, um computador transmite um sinal que todos os outros computadores acoplados ao barramento recebem. Uma topologia em anel consiste em computadores conectados em um loop fechado. O primeiro computador está conectado ao segundo, o segundo ao terceiro, e assim por diante, até o último computador, que está conectado ao primeiro. Finalmente, uma topologia em estrela se assemelha a uma roda, com a rede propriamente dita correspondendo a um hub central, e os links para computadores individuais correspondendo aos aros. Cada topologia tem vantagens e desvantagens; nenhuma topologia é melhor em todos os propósitos.

Existem tecnologias de LAN que usam cada uma dessas topologias. Uma LAN Ethernet usa uma topologia de barramento, assim como o LocalTalk. Para acessar uma Ethernet, as estações obedecem ao Carrier Sense Multiple Access with Collision Detection (CSMA/CD). Isto é, uma estação espera que o éter esteja inativo e então tenta enviar. Se duas estações transmitem ao mesmo tempo, há uma colisão, fazendo com que elas esperem um tempo aleatório antes de tentar novamente. Sucessivas colisões causam backoff exponencial através do qual cada estação dobra seu atraso.

LANs sem fio como as redes 802.11 usam Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). Antes de transmitir um quadro de dados, um remetente transmite uma pequena mensagem de controle à qual o receptor responde. A troca de mensagens de controle notifica todas as estações dentro do alcance do receptor que uma transmissão de dados está para acontecer. Outras estações então permanecem silenciosas enquanto a transmissão acontece (ou seja, evitam uma colisão), ainda que não recebam uma cópia do sinal.

As estações acopladas a uma rede em anel com passagem de token também compartilham o meio. Enquanto uma estação transmite um quadro, todas as demais passam os bits em torno do anel, que permite ao remetente verificar se os bits foram corretamente transmitidos. Para coordenar o uso do anel e garantir justiça no acesso, as estações em um token ring enviam uma mensagem especial chamada token. Uma estação espera que o token chegue, usa o anel completamente para transmitir um quadro e então envia o token para a próxima estação. Tecnologias mais antigas, como o Token Ring da IBM e as redes FDDI fazem uso da passagem de token. Um anel de passagem de token pode usar um anel extra para se recuperar de falhas catastróficas. O anel extra é chamado de rotação contrária porque os dados fluem na direção oposta à do anel principal. Diz-se que uma rede com um anel de rotação contrária é de autocura (self-healing) porque pode detectar uma falha e fazer o loop ao longo do anel reverso para fechar o caminho.

As LANs que usam a tecnologia ATM tem uma topologia em estrela. Um switch ATM forma o hub da estrela ao qual cada computador se conecta. Já que o ATM é projetado para operar em alta velocidade, a conexão entre um computador e um switch ATM usa um par de fibras ópticas, com cada fibra transportando dados em uma direção.

Exercícios

- 8.1 Mantenha um registro das pessoas com quem você troca correio eletrônico (e-mail) por uma semana e da hora em que as mensagens são enviadas. Sua comunicação exibe localidade espacial de referência? Exibe localidade temporal de referência?
- 8.2 Veja se você pode encontrar um exemplo de uma rede ponto a ponto. Pergunte ao proprietário por que foram escolhidas conexões ponto a ponto.
- 8.3 Classifique a topologia de cada rede em seu site. Qual topologia é a mais usada? Qual é a menos usada?
- 8.4 Um cabo de Ethernet deve ter um terminador em cada extremidade para prevenir reflexões. Consulte uma fonte sobre engenharia elétrica para descobrir que tipo de componente é usado em um terminador.

- 8.5** Assuma que um arquivo de um megabyte deve ser transferido através de uma rede. Ignorando atrasos causados por espera no acesso e outras sobrecargas (por exemplo, contando somente os dados transferidos), quanto tempo levaria para enviar o arquivo através de uma Ethernet? E através de uma rede LocalTalk? E através de uma Fast Ethernet ou uma rede FDDI?
- 8.6** O padrão Ethernet especifica um tamanho de quadro mínimo, como também um tamanho máximo. Fale com um engenheiro elétrico para descobrir por que é necessário um tamanho mínimo. (Dica: como uma mensagem de 1 bit aparece para o hardware?)

Sumário do Capítulo

- | | | |
|-------------|---|-----|
| 9.1 | Introdução | 133 |
| 9.2 | Especificando um Receptor | 133 |
| 9.3 | Como o Hardware de LAN Usa Endereços para Filtrar Pacotes | 134 |
| 9.4 | Formato de um Endereço Físico | 135 |
| 9.5 | Broadcasting | 137 |
| 9.6 | Multicasting | 137 |
| 9.7 | Endereçamento Multicast | 138 |
| 9.8 | Identificando o Conteúdo de Pacotes | 139 |
| 9.9 | Cabeçalhos e Formatos de Quadro | 139 |
| 9.10 | Um Exemplo de Formato de Quadro | 140 |
| 9.11 | Usando Redes que Não Têm Quadros Auto-Identificados | 140 |
| 9.12 | Analisadores de Rede, Endereços Físicos, Tipos de Quadro | 143 |
| 9.13 | Resumo | 144 |



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

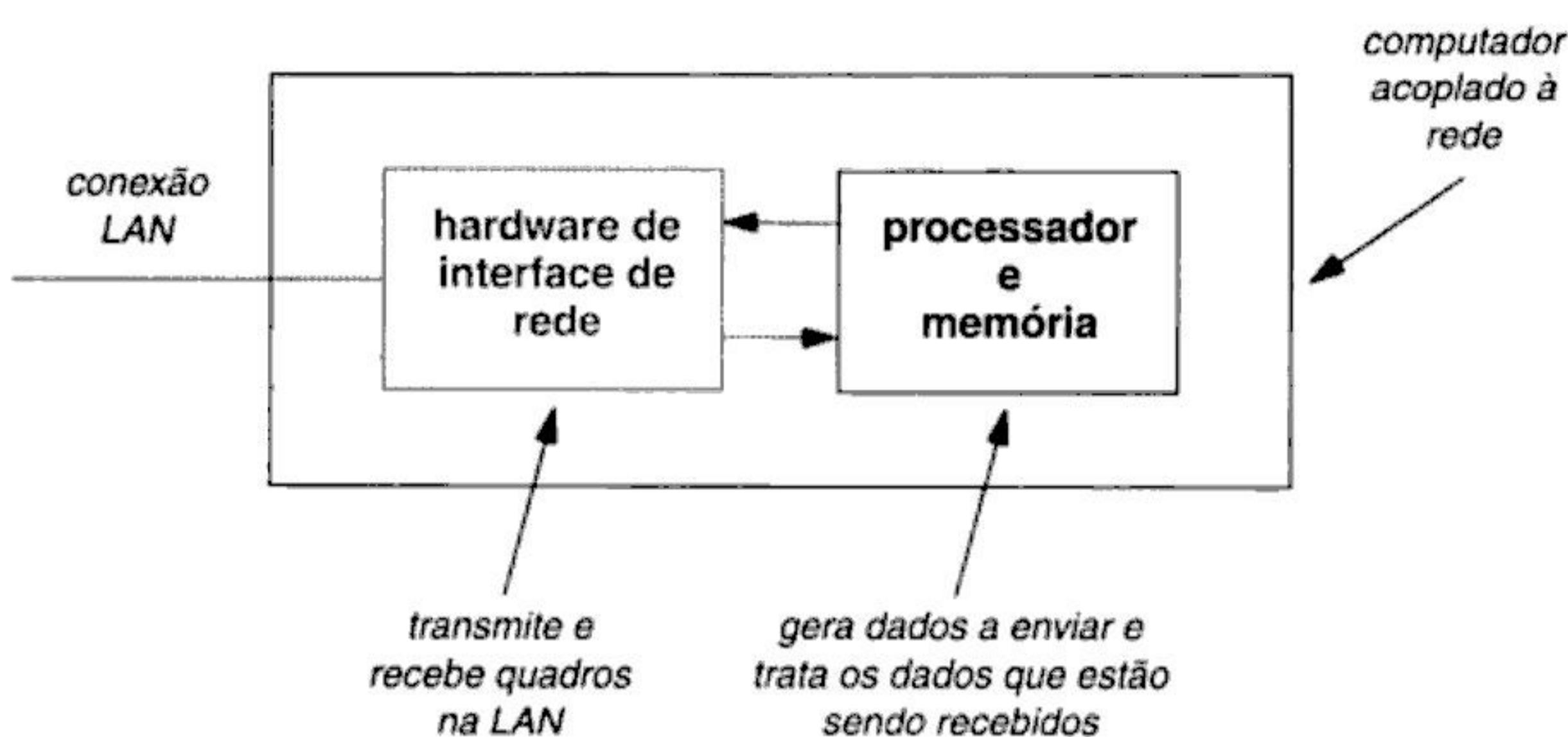


Figura 9.1 A organização do hardware em um computador acoplado a uma LAN. Como é poderoso e independente, o hardware de interface de rede não usa a CPU ao transmitir ou receber bits de um quadro.

tes que viajam através da LAN. Lembre-se de que, como um quadro viaja através do meio compartilhado, uma cópia do sinal passa em cada estação. Uma vez capturado um quadro completo, o hardware de interface compara o endereço de destino no quadro com o endereço físico da estação. Se o endereço de destino combina com o endereço físico da estação, o hardware aceita o quadro e passa o mesmo para o sistema operacional. Se o endereço de destino no quadro não combina com o endereço físico da estação, o hardware descarta o quadro e espera pelo próximo quadro; um quadro endereçado a uma estação inexistente é ignorado. Já que o hardware de interface de LAN funciona sem usar o processador central do computador, a captura e a comparação de endereço não interferem com a computação normal. Deste modo, um computador em uma LAN compartilhada permanece isolado da maior parte das atividades na rede – do ponto de vista do computador, uma interface de LAN faz somente uma cópia dos quadros que são destinados a ele.

Para resumir:

Um sistema de rede compartilhada usa endereços físicos para filtrar quadros entrantes. O hardware de interface de rede, que trata de todos os detalhes de transmissão e recepção de quadros, compara o endereço de destino em cada quadro recebido com o endereço físico da estação e descarta os quadros não destinados à estação. Como uma interface de rede opera sem usar a CPU de uma estação, um quadro pode ser transferido através de uma LAN compartilhada de um computador a outro sem interferir com o processamento em outros computadores.

9.4 Formato de um endereço físico

Várias perguntas permanecem sem respostas. Que valores numéricos são usados para endereços físicos? Como são atribuídos tais endereços? Onde o endereço de destino e os endereços de origem são localizados em um quadro? As respostas a essas perguntas dependem da tecnologia de LAN que está sendo usada. Por exemplo, a Ethernet usa uma forma de endereçamento, enquanto uma tecnologia como a FDDI usa outra. As várias formas de endereçamento podem ser agrupadas em três grandes categorias:

Estático

Um *esquema de endereçamento estático* se baseia no fabricante de hardware para designar um endereço físico único para cada interface de rede. Um endereço físico estático não muda, a menos que o hardware seja substituído.

Configurável

Um *esquema de endereçamento configurável* fornece um mecanismo que um cliente pode usar para configurar um endereço físico. O mecanismo pode ser manual (por exemplo, switches que devem ser configurados quando a interface é instalada pela primeira vez) ou eletrônica (por exemplo, uma memória não-volátil, como uma EPROM, que pode ser baixada de um computador). A maioria dos hardwares precisa ser configurada somente uma vez – a configuração é normalmente feita quando o hardware é instalado pela primeira vez.

Dinâmico

Um *esquema de endereçamento dinâmico* fornece um mecanismo que designa automaticamente um endereço físico a uma estação quando a estação é inicializada (boot) pela primeira vez. A maioria dos esquemas de endereçamento dinâmicos exige que uma estação tente números aleatórios até que encontre um valor que nenhum outro computador esteja usando como endereço. Por exemplo, uma estação poderia escolher a hora atual do dia como um valor inicial. Para cada número aleatório gerado, a estação envia uma mensagem através da rede para o endereço especificado. Se outro computador já estiver usando o endereço, o computador responde à mensagem. Se nenhuma outra estação responder para um dado endereço, o remetente pode usá-lo como seu endereço físico. Deste modo, o endereço que um computador seleciona depende dos endereços que os outros computadores estão usando quando ele é inicializado – um computador poderia obter um endereço diferente cada vez que é reiniciado.

As vantagens principais de endereçamento estático são a facilidade de uso e a permanência. O esquema é fácil de usar porque os vendedores de hardware designam endereços e asseguram que cada dispositivo de hardware tem um endereço físico único em todo o mundo. Os dispositivos de hardware de múltiplos fabricantes podem ser conectados a uma rede física única sem conflitos de endereço. O endereçamento estático é permanente porque o endereço de um computador não muda quando o computador é reinicializado.

O endereçamento dinâmico tem duas vantagens: elimina a necessidade de os fabricantes de hardware se coordenarem ao atribuir endereços e permite que cada endereço seja menor. O que torna possível que os endereços sejam menores é que a unicidade é importante apenas dentro de uma única LAN. Um esquema de endereçamento dinâmico permite às estações em uma LAN escolherem os mesmos endereços que as estações em outras LANs. As principais desvantagens do endereçamento dinâmico são a falta de permanência e o conflito potencial. Cada vez que um computador é reinicializado, ele obtém um endereço novo; outros computadores devem ser informados do novo endereço antes de poderem se comunicar. Além disso, se a rede for temporariamente desconectada durante uma inicialização de um computador, dois computadores podem escolher o mesmo endereço físico.

Os endereços configuráveis fornecem um meio-termo entre os esquemas estáticos e dinâmicos. Como os endereços estáticos, os endereços configuráveis são permanentes – continuam os mesmos ao longo das reinicializações. Como os endereços dinâmicos, os configuráveis não precisam ser grandes porque o endereço é único somente em uma dada rede. Na prática, a maioria dos administradores de rede escolhe designar a endereços configuráveis valores seqüenciais. Ao primeiro computador inserido em uma rede é atribuído o endereço 1, ao segundo é atribuído o endereço 2, e assim por diante. Uma das vantagens de um esquema de endereçamento configurável se torna aparente quando o hardware de interface de rede falha e deve ser substituído: diferentemente do hardware que usa uma designação estática, uma interface configurável pode ser substituída sem mudar o endereço físico do computador.

9.5 Broadcasting

Muitos aplicativos que usam uma rede se baseiam em uma técnica conhecida como *broadcasting* (ou *difusão*). O termo, originalmente aplicado a transmissões de rádio e televisão, refere-se a transmissões disponíveis a um grande público. Quando um aplicativo faz broadcast de dados, ele torna uma cópia dos mesmos disponível a todos os outros computadores na rede.

O broadcast tem muitos usos. Por exemplo, suponha que um computador precise achar uma das impressoras na rede. Ele pode formar uma mensagem que especifica uma impressora (por exemplo, dando o nome dela) e então enviar por broadcast a mensagem para todas as estações na rede. Embora cada estação vá receber o broadcast, somente a impressora nomeada responderá.

Como as tecnologias de Rede Local empregam, em sua maioria, meios compartilhados, elas podem realizar broadcast de forma extremamente eficiente. Nenhum hardware adicional é necessário para o broadcast em uma LAN, pois todas as estações se conectam diretamente ao meio compartilhado. Deste modo, todas as estações recebem uma cópia do sinal cada vez que um quadro é transmitido através da rede. Tudo que é necessário para possibilitar um broadcast eficiente é um mecanismo que faça com que todas as estações extraiam e processem uma cópia do quadro.

O endereçamento físico descrito parece tornar o envio por broadcast impossível. Em particular, embora todas as estações recebam uma cópia de cada quadro, o hardware de interface em cada uma usa o endereço de destino do quadro para determinar se deve manter uma cópia. Se o campo de endereço de destino no quadro contiver o endereço de um computador específico, *C*, somente o hardware de interface de rede do computador *C* aceitará tal quadro; todas as outras interfaces de rede o rejeitarão.

Para tornar o uso de broadcast eficiente, a maioria das tecnologias de LAN estende o esquema de endereçamento. Além de designar a cada computador um endereço, os projetistas de rede definem um endereço especial reservado, conhecido como *endereço de broadcast*. A interface de hardware em um computador é construída para reconhecer o endereço especial de broadcast e também o endereço físico da estação³. Se um quadro chega com qualquer um dos dois endereços, a interface o aceita e entrega uma cópia para o sistema operacional do computador. Para resumir:

O hardware de interface de rede em um computador faz uma cópia de cada quadro que passa através da rede compartilhada. A interface aceita o quadro e entrega uma cópia para o sistema operacional se o endereço do destino no quadro for o endereço de broadcast ou o endereço físico da estação. Então, quando um quadro é enviado para um endereço de broadcast, cada computador na rede recebe uma cópia.

9.6 Multicasting

Em teoria, o broadcasting pode ser usado para uma variedade de aplicativos. Tome uma impressora de rede, por exemplo. Suponha que cada vez que um aplicativo necessitasse usar uma impressora, ele usasse a rede para enviar, via broadcast, os quadros contendo os dados a serem impressos. Como os quadros de broadcast alcançam todas as estações na rede, uma cópia alcançaria a impressora. As outras estações que recebessem uma cópia de quadros contendo dados para impressão poderiam ser configuradas para descartá-los, fazendo com que o esquema inteiro funcionasse corretamente.

Apesar da viabilidade aparente de usar broadcast, poucos aplicativos de rede foram projetados para usá-lo como o descrito. A razão é simples – enviar por broadcast é extremamente ineficiente. Embora cada estação em uma rede possa ser configurada para descartar os quadros desnecessá-

³ Como representa uma única estação, o endereço físico convencional é classificado como endereço *unicast*.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

9.10 Um exemplo de formato de quadro

Um exemplo elucidará o conceito de um formato de quadro e mostrará como os campos de endereços e tipo aparecem em um cabeçalho de quadro. A Figura 9.3 ilustra o formato de quadro usado em uma Ethernet. Conforme a figura, o quadro de Ethernet ilustrado começa com um cabeçalho que contém três campos. O *preâmbulo* de 64 bits que precede o quadro contém *1s* e *0s* alternados, que permitem ao hardware do receptor sincronizar-se com o sinal que está sendo recebido. Os dois primeiros campos do cabeçalho contêm endereços físicos. A Ethernet usa um esquema de endereçamento estático de 48 bits, em que cada dispositivo é designado pelo fabricante com um endereço único. O campo de nome *Dest. Address* contém o endereço físico da estação para a qual o quadro está sendo enviado. O campo de nome *Source Address* contém o endereço físico da estação que enviou o quadro. O terceiro campo do cabeçalho consiste em um tipo de quadro (*frame type*) Ethernet de 16 bits.

O padrão Ethernet Intel-Digital-Xerox especifica os valores que podem ser usados nos campos de cabeçalho e seus significados. Por exemplo, o padrão especifica que o endereço com todos os 48 bits configurados em *1* é reservado para broadcast, outros endereços que começem com um bit *1* são usados para transmissões multicast e o valor hexadecimal *8137* no campo de *Frame Type* especifica que os dados nos quadros seguem um protocolo da Novell Corporation conhecido como *IPX*. Centenas de valores de tipo para Ethernet foram designados; a tabela na Figura 9.4 contém alguns exemplos.

Como mostra a figura, tipos de Ethernet foram designados para uso com sistemas construídos por empresas individuais, bem como para uso com software que segue os padrões internacionais, como o X.25. Os tipos padronizados asseguram que todos os produtos de Ethernet usam o mesmo valor para um tipo determinado de quadro. Deste modo, os produtos de Ethernet construídos por dois ou mais vendedores interoperarão. Além disso, a Seção 9.12 descreve como podem ser usados tipos padronizados de quadro para analisar uma rede.

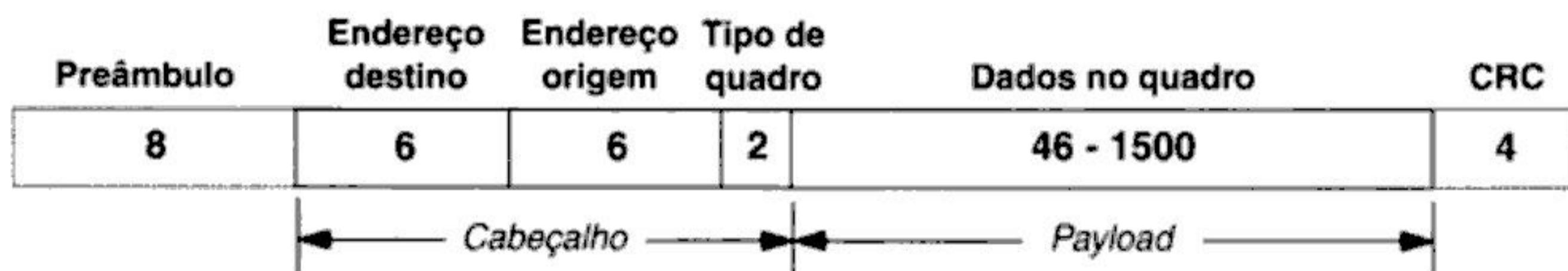


Figura 9.3 Ilustração do formato de quadro usado com Ethernet. O número em cada campo indica o tamanho do campo medido em octetos de 8 bits.

9.11 Usando redes que não têm quadros auto-identificados

Algumas tecnologias de rede não incluem um campo de tipo no cabeçalho de quadro. Isto é, os quadros não são auto-identificados. Como os computadores conectados a tais redes podem saber o tipo de dados de cada quadro? Existem duas abordagens possíveis:

- Antes de qualquer dado ser enviado, remetente e receptor concordam em usar um formato único. O software no computador remetente é programado para colocar os dados que estão sendo enviados no formato escolhido, e o software no computador receptor é programado para esperar os dados naquele formato.
- Antes de qualquer dado ser enviado, remetente e receptor concordam em usar os primeiros octetos do campo de dados para armazenar informações de tipo. O software no computador reme-

Valor	Significado
0000-05DC	Reservado para uso com IEEE LLC/SNAP
0800	Internet IP Versão 4
0805	X.25 de CCITT
0900	Depurador de rede da Ungermann-Bass Corporation
OBAD	VINES da Banyan Systems Corporation
1000-100F	Encapsulamento de cauda do Berkeley UNIX
6004	LAT da Digital Equipment Corporation
6559	Frame Relay
8005	Probe de rede da Hewlett Packard Corporation
8008	AT&T Corporation
8014	Jogos de rede da Silicon Graphics Corporation
8035	ARP da Reverse Internet
8038	LANBridge da Digital Equipment Corporation
805C	V Kernel da Stanford University
809B	AppleTalk da Apple Computer Corporation
80C4-80C5	Banyan Systems Corporation
80D5	SNA da IBM Corporation
80FF-8103	Wellfleet Communications
8137-8138	Novell Corporation IPX
818D	Motorola Corporation
FFFF	Reservado

Figura 9.4 Exemplos de tipos de quadro usados com Ethernet (valores de tipo são fornecidos em hexadecimal). A tabela lista somente alguns exemplos; muitos outros tipos tem sido determinados.

tente acrescenta informações de tipo ao início dos dados, antes de colocá-los em um quadro que está sendo enviado. O software no computador receptor extrai as informações de tipo e usa o tipo para determinar como processar o restante dos dados.

A primeira técnica raramente é usada, pois limita um par de computadores a apenas um formato de dados. Desse modo, os proprietários dos dois computadores não podem instalar um software aplicativo novo, a menos que os aplicativos usem o formato de dados selecionado. Mais importante, se a rede suporta broadcast, todos os computadores acoplados a ela devem concordar em usar um formato de dados único.

A Figura 9.5 mostra como a segunda técnica usa parte da área de dados para armazenar informações de tipo.

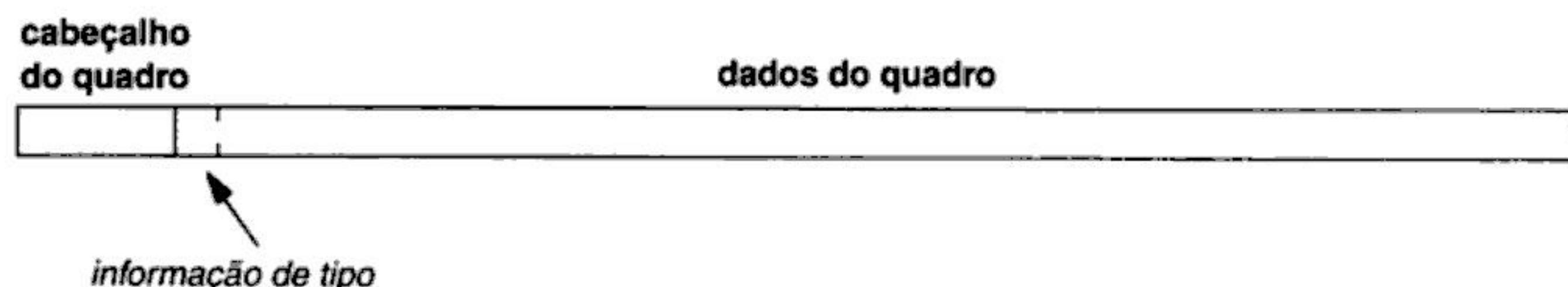


Figura 9.5 Ilustração de como podem ser incluídas informações de tipo na área de dados de um quadro se o cabeçalho não inclui um campo de tipo.

O uso de parte da área de dados para transportar o tipo de quadro dá margem a duas perguntas. Primeiro, exatamente de que tamanho deveriam ser as informações de tipo? Segundo, quem deveria especificar os valores permitidos no campo de tipo e o seu significado? Para as redes que incluem um campo de tipo no cabeçalho de quadro, tais decisões são feitas pelo grupo que projeta a tecnologia de hardware. Porém, se o hardware não inclui um campo de tipo de quadro, o software é livre para escolher como interpretar as informações de tipo. Infelizmente, permitir que cada programador de aplicativo escolha valores para o campo de tipo não funciona bem já que dois programadores poderiam selecionar accidentalmente o mesmo valor para tipos diferentes. O problema é especialmente grave se os aplicativos enviam quadros via broadcast.

Para assegurar que todo software concorda com os valores usados para especificar tipos, as organizações padronizadoras definiram o significado de cada valor. Infelizmente, muitas organizações padronizadoras fazem tais tarefas, e nem sempre elas coordenam seus esforços. Como resultado, duas organizações podem designar accidentalmente o mesmo valor para dois tipos diferentes. Para resolver o problema, o IEEE definiu um padrão que inclui um campo para identificar a organização padronizadora e outro para identificar um tipo tal como definido por aquela organização. Parte do padrão do IEEE, a especificação é conhecida como um cabeçalho *Ponto de Acoplamento de Sub-rede* (*SubNetwork Attachment Point, SNAP*) para Controle de Link Lógico (*Logical Link Control, LLC*). O cabeçalho IEEE LLC/SNAP é amplamente aceito.

A Figura 9.6 apresenta um exemplo de cabeçalho LLC/SNAP que contém oito bits. Os primeiros três incluem a porção de LLC, que especifica que a seguir há um campo de tipo.

Como a figura mostra, a porção SNAP do cabeçalho é dividida em dois campos. O primeiro é chamado de *Identificador de Organização Única* (*Organizationally Unique Identifier, OUI*), usado para identificar uma organização padronizada em particular. O segundo contém um valor de *tipo* (*type*) conforme definido por aquela organização. Por exemplo, o valor de OUI de todos os zeros mostrados na Figura 9.6 é designado para a organização que especifica tipos para Ethernet. Desse modo, o valor hexadecimal 0800 mostrado no campo *TYPE* do exemplo é interpretado de acordo com o padrão de tipos para Ethernet.

Como um tipo codificado em um cabeçalho de quadro, o campo de tipo LLC/SNAP possibilita que todos os computadores em uma rede compartilhada enviem quadros via broadcast. Quando um quadro chega a um computador, este procura por informações de LLC/SNAP no princípio da área de dados do quadro. Se o receptor não reconhece o OUI ou não tem software para tratar o tipo de dados que estão sendo enviados o quadro é descartado. Deste modo, se apenas três computadores em uma rede entendem um determinado tipo, um quadro de broadcast que está transportando dados daquele tipo será ignorado por todos os computadores, exceto aqueles três.

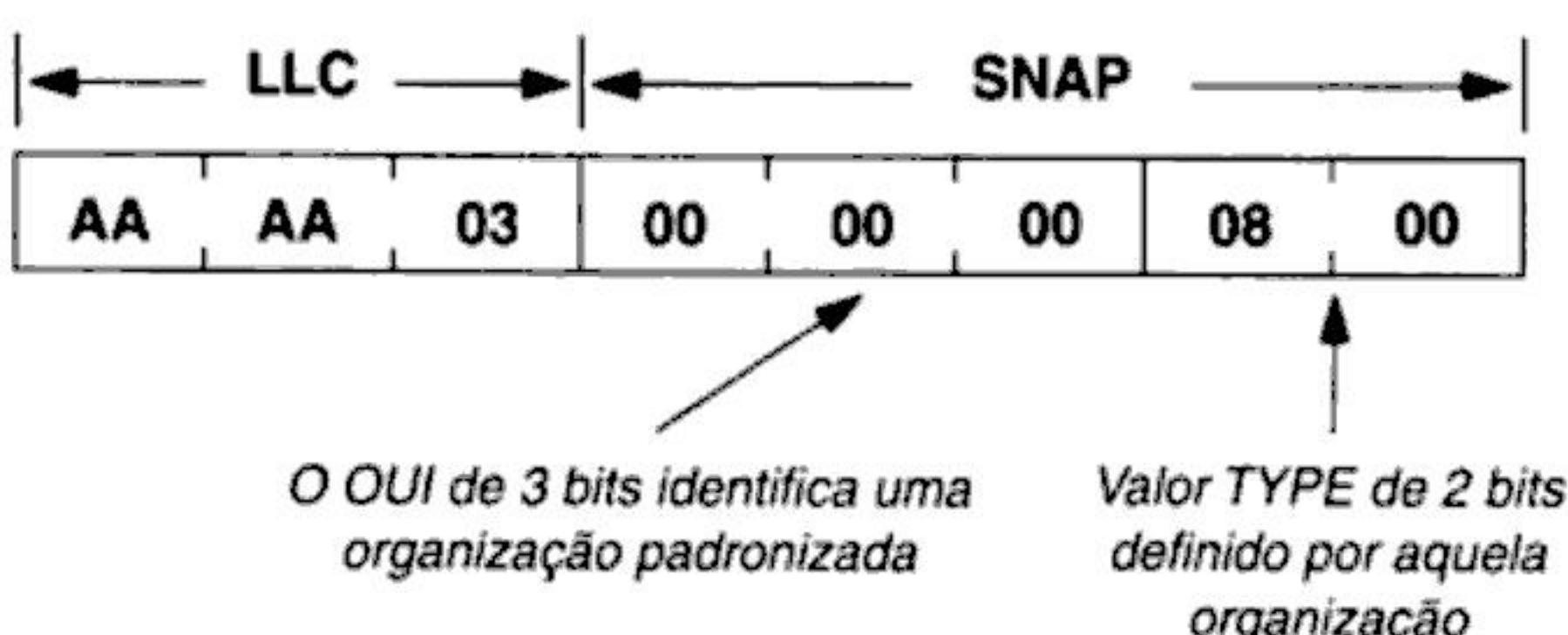


Figura 9.6 Um exemplo do cabeçalho IEEE de LLC/SNAP com 8 bits, usado para especificar o tipo de dados. A porção SNAP especifica uma organização e um tipo definido pela mesma.

9.12 Analisadores de rede, endereços físicos, tipos de quadro

Um *analisador de rede* ou *monitor de rede* é um dispositivo usado para determinar quão bem um sistema de rede está executando⁵. A maioria dos analisadores são portáteis, podendo ser movidos facilmente. Após ser acoplado a uma rede, um analisador pode monitorar eventos específicos e relatar estatísticas como o número médio de quadros por segundo ou o tamanho médio de quadro. Por exemplo, um analisador projetado para uma rede de CSMA/CD (como uma Ethernet) pode relatar o número de colisões que acontecem. Um analisador projetado para ser usado com uma rede token ring pode relatar o atraso médio até a chegada do token ou o número de estações que transmitem em um determinado ciclo do token. Mais importante, a maioria dos analisadores é flexível – um gerente pode configurá-lo para observar os quadros enviados por uma máquina específica, observar tráfego de um tipo específico ou computar a porcentagem de quadros de cada tipo.

Como trabalha um analisador de rede? O hardware necessário para um analisador é surpreendentemente simples – muitos analisadores consistem em um computador de padrão portátil (por exemplo, um PC notebook) com uma interface padrão de LAN. Enquanto usado como analisador, o computador é completamente dedicado à tarefa de análise. O software necessário também é simples. O programa analisador começa permitindo que um usuário configure parâmetros, e então usa os parâmetros para analisar pacotes.

Para ler pacotes, o software analisador coloca o hardware de interface de rede do computador em *modo promíscuo*, que anula o reconhecimento de endereço convencional. Isto é, o software configura a interface de rede do computador para aceitar *todos* os quadros. Uma vez em modo promíscuo, a interface não verifica o endereço de destino nem rejeita quaisquer quadros. Em vez disso, ela simplesmente coloca uma cópia de cada quadro na memória do computador e interrompe a CPU para informá-la de que chegou um quadro.

Os usuários que não são informados podem assumir que uma placa de interface especial é exigida para modo promíscuo. Surpreendentemente, este não é o caso. Quase todo hardware de interface de rede comercialmente disponível suporta leitura promíscua. Além disso, colocar uma interface em modo promíscuo é normalmente trivial – apenas uma grande quantidade de instruções de CPU precisa ser executada. Como consequência, qualquer usuário com um computador acoplado a uma LAN pode ler todos os pacotes que viajam pela LAN. Isto é, qualquer computador acoplado a uma LAN pode espiar toda comunicação; as mensagens enviadas através de uma LAN *não* têm garantia de privacidade.

Para entender o que é possível, considere como trabalha um analisador de rede. Cada vez que um quadro viaja através da rede, o hardware de interface de rede faz uma cópia e a entrega para o programa analisador. Este pode examinar campos no cabeçalho do quadro para determinar o remetente, receptor pretendido, tipo, ou pode examinar a área de dados.

Um analisador de rede é configurável; a configuração exata que um usuário seleciona determina quais campos o analisador examina e que informações ele mantém. Por exemplo, suponha que um administrador de rede esteja tentando depurar um problema em que um computador parece estar emitindo pacotes incorretos. Se o usuário sabe o endereço físico do computador, P , ele pode configurar um analisador para examinar todos os quadros que se originam do computador em questão. Para fazer tal pedido, o administrador especifica que o analisador deveria mostrar todos os quadros com endereço de origem igual a P . Ao receber cada quadro, o analisador comparará o campo de endereço de origem no cabeçalho do quadro com P . Quando encontra uma igualdade, o analisador apresenta o quadro. Caso contrário, ele ignora o quadro e segue para o próximo. Semelhante-

⁵ Informalmente, um analisador de rede é às vezes chamado de sniffer de rede, devido ao popular produto da Network General Corporation.

mente, se um administrador suspeita de um problema com tráfego de broadcast, ele pode configurar um analisador para relatar todos os quadros de broadcast.

Como alternativa, um administrador poderia especificar que o analisador deve juntar estatísticas sobre tráfego em geral. Para fornecer um resumo continuamente atualizado de tais informações, um analisador mantém uma estrutura de dados que possui contadores para cada tipo de quadro possível. Os contadores começam em zero. Quando um quadro chega, o analisador extrai o valor do campo de tipo de quadro no cabeçalho e usa esse valor para determinar qual contador deve ser incrementado. Periodicamente, o programa analisador atualiza a tela do usuário. Ele computa a porcentagem de quadros totais de cada tipo e apresenta o relatório em um formato conveniente (por exemplo, um gráfico de setores circulares). O analisador continua também a acumular contadores. Normalmente, o intervalo de tempo de atualização escolhido é pequeno (por exemplo, um segundo ou menos) para dar a impressão de que a tela é continuamente atualizada.

Um analisador de rede é um dispositivo que pode ser configurado para contar ou apresentar quadros à medida que estes passam através de uma rede compartilhada. Um analisador obtém uma cópia de cada quadro e então usa os campos de cabeçalho, como o endereço de origem física, o endereço de destino físico, ou as informações de tipo, para determinar como processar o quadro.

9.13 Resumo

Embora as redes de computadores possam ser compartilhadas por muitos computadores, um esquema de endereçamento torna possível enviar um pacote a um computador específico. Cada computador acoplado à rede compartilhada é designado com um número único, conhecido como o endereço físico do computador. Todos os computadores acoplados à rede recebem uma cópia dos sinais elétricos que se propagam através do meio compartilhado; um computador ignora qualquer quadro endereçado a outros computadores.

O hardware de interface de rede trata dos detalhes de transmissão e recepção de quadros, além da checagem do endereço. Uma interface checa o endereço de destino no cabeçalho de cada quadro e compara com o endereço físico do computador e o endereço de broadcast. Se o endereço de destino no quadro bate com os da estação, o hardware de interface interrompe a CPU e passa o quadro recebido para o computador para processamento. Caso contrário, o computador descarta o quadro.

O broadcasting é eficiente em uma Rede Local compartilhada porque um pacote de broadcast precisa ser transmitido apenas uma vez; todos os computadores recebem uma cópia da transmissão. O multicasting é uma forma limitada de broadcast, com a vantagem de usar o hardware de interface de rede para examinar quadros. Cada computador configura sua interface para aceitar quadros destinados a um dado endereço multicast.

Os detalhes exatos de endereços físicos dependem da tecnologia. Endereçamento estático requer que o fabricante do hardware de interface atribua a cada unidade um endereço único. Endereços configuráveis requerem que um administrador do sistema escolha um endereço único para cada computador; endereçamento dinâmico usa um software que escolhe um endereço físico quando o computador é ligado.

O cabeçalho do quadro pode incluir um campo usado para especificar o tipo de quadro. Um tipo único deve ser atribuído a cada uso possível da rede. O remetente coloca o valor correto no campo de tipo para especificar o tipo de dados nos quadros; um receptor examina o conteúdo do campo de tipo para determinar como tratar o quadro.

Um analisador de rede é um dispositivo que pode ser usado para depurar problemas em uma rede. Ele recebe uma cópia de cada quadro que passa através da rede compartilhada e processa estatísticas na forma de uma soma para cada tipo de quadro.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

Cabeamento de LAN, Hardware de Topologia e Interface Físicos

10.1 Introdução

Este capítulo continua a discussão sobre Redes Locais examinando os esquemas de cabeamento de hardware em mais detalhes. O capítulo começa fazendo algumas considerações sobre placas de interface de rede que conectam um computador a uma rede e trata dos detalhes de transmissão e recepção de pacotes. Discute ainda a motivação para se ter uma placa de interface dedicada, descreve como uma placa se conecta a um computador com outros dispositivos de E/S e explica como ela usa um transceiver para interagir com o meio de rede.

Depois de discutir as interfaces, o capítulo descreve o cabeamento de LAN e identifica os componentes de hardware importantes usados em vários esquemas de cabeamento, incluindo uma descrição de hubs. Finalmente, discute a diferença entre topologia lógica e topologia física, e mostra como os conceitos discutidos no Capítulo 7 são implementados na prática.

10.2 Velocidades de LANs e computadores

Cada tecnologia de rede especifica uma taxa em que os dados devem ser enviados. Surpreendentemente, muitas redes locais operam a uma taxa tão rápida que a Unidade Central de Processamento (*Central Processing Unit, CPU*) de um computador não consegue processar os bits na velocidade da rede. Por exemplo, o Capítulo 7 relata que uma rede de Ethernet pode operar a 1 Gigabite por segundo. Pelos padrões atuais, uma CPU razoável usa um clock de hardware que opera a 3 GigaHertz¹. Ainda que possa executar uma instrução em cada ciclo de clock, uma CPU que opera a 3 GigaHertz não pode tratar de dados tão rapidamente quanto chegam de uma rede de Ethernet porque são exigidas várias instruções para cada bit.

A diferença entre a velocidade de uma CPU e a velocidade de uma rede é um problema fundamental. Não faz sentido operar uma rede a uma velocidade adequada à CPU mais lenta, porque dessa forma também é reduzida a velocidade de transferência de dados entre um par de computadores de

¹ Abreviado como GHz, um GigaHertz corresponde a um bilhão de ciclos por segundo; outra medida comum é o MegaHertz, abreviado como MHz, que um corresponde a um milhão de ciclos por segundo.

alta velocidade. Além disso, não faz sentido especificar que todos os computadores acoplados a uma rede devem operar na mesma velocidade, pois os projetistas continuam a inventar processadores. A melhoria ininterrupta na velocidade dos processadores disponíveis significa que sempre que um site substitui um computador antigo, o substituto será significativamente mais rápido que o original. Como resultado, os computadores acoplados a uma rede típica não operam todos na mesma velocidade.

Apesar da diferença entre as velocidades do processador e da rede, as redes são projetadas para operar na taxa mais alta que o hardware pode suportar. Além disso, a velocidade em que uma rede opera é normalmente fixa durante o projeto – a velocidade não depende das taxas de CPU dos computadores acoplados.

10.3 Hardware de interface de rede

Como um computador pode se ligar a uma rede que envia e recebe bits mais rapidamente do que a CPU do computador pode tratá-los? A resposta é simples: a CPU não trata da transmissão ou da recepção de bits individuais. Em vez disso, um componente de hardware de propósito específico conecta um computador a uma rede e trata de todos os detalhes de transmissão e recepção de pacotes. Fisicamente, o hardware de propósito específico normalmente consiste em uma placa de circuitos impressos que contém componentes eletrônicos. Conhecida como *placa adaptadora de rede* ou *placa de interface de rede (network interface card, NIC)*, a placa de circuitos impressos se conecta ao barramento do computador, e um cabo a conecta ao meio da rede. A Figura 10.1 mostra como os sockets usados nas placas de interface poderiam ser posicionados em um computador.

Como mostra a figura, os sockets para as placas de interface estão normalmente localizados próximos à parte de trás do gabinete. Cada NIC é instalada verticalmente em um socket, com um lado exposto através de uma abertura na parte de trás do gabinete. O lado exposto contém um conector

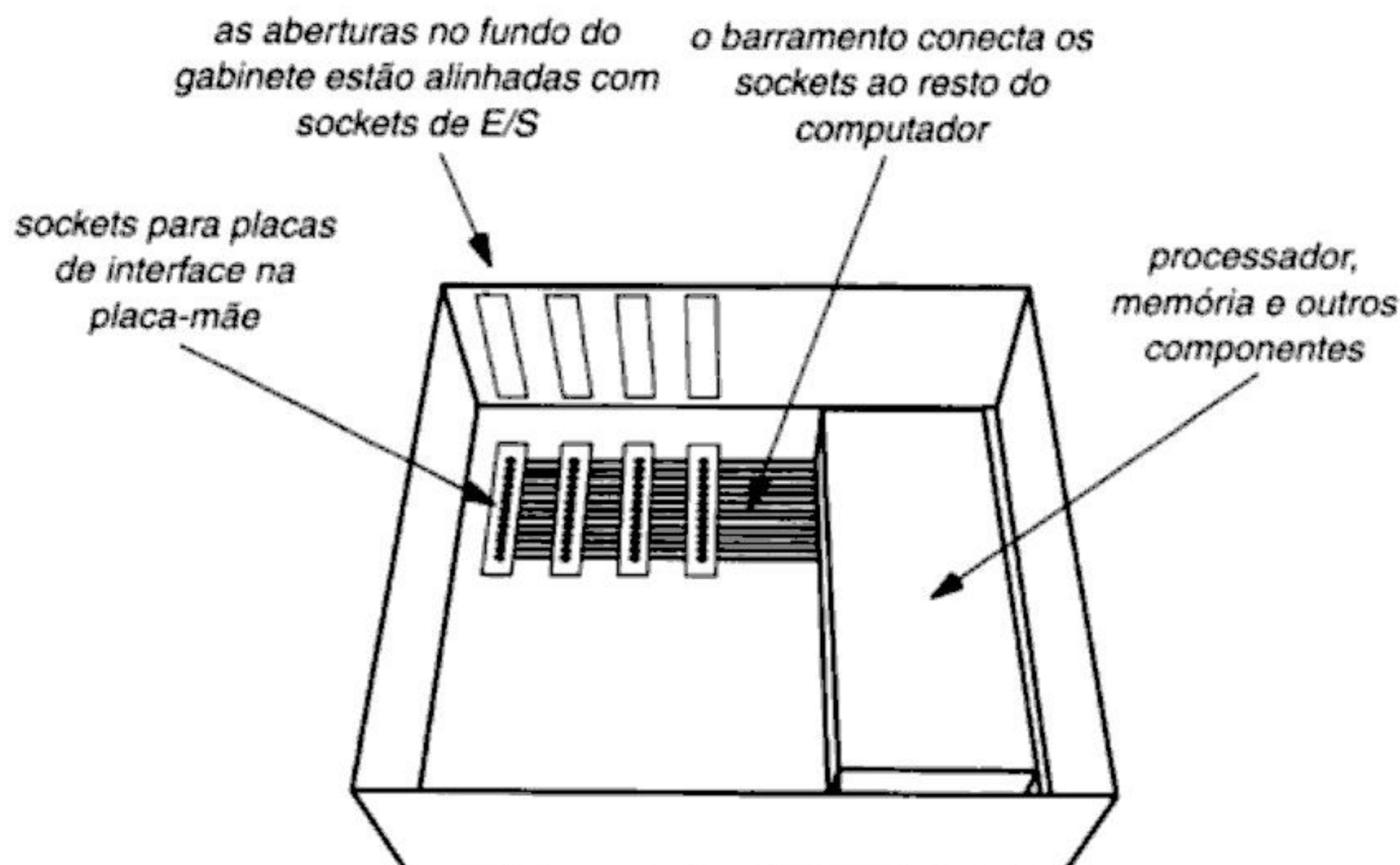


Figura 10.1 A localização de sockets de E/S dentro de um computador típico. Cada socket se alinha com uma fenda atrás do gabinete, e o barramento do computador conecta o socket a outros componentes importantes, como o processador e a memória.

– um cabo liga o conector à rede. A Figura 10.2 mostra como a conexão na parte de trás de uma NIC poderia aparecer.

Uma NIC entende os sinais elétricos usados em uma rede, a taxa em que os dados devem ser enviados ou recebidos e os detalhes do formato de quadro da rede. Por exemplo, uma NIC projetada para ser usada com Ethernet não pode ser usada em outros tipo de redes.

A maior parte das NICs contém circuitos de *Acesso Direto à Memória* (*Direct Memory Access, DMA*), que permitem à NIC operar independentemente da CPU – a NIC pode transmitir ou receber dados da memória sem usar a CPU do computador.² Do ponto de vista da CPU, a NIC parece operar como qualquer dispositivo de E/S (p.ex., como um disco). Para transmitir na rede, a CPU forma um pacote na memória e então instrui a NIC para começar a transmitir. A CPU pode continuar outras tarefas enquanto a NIC lida com os detalhes do acesso ao meio e da transmissão de bits (assim como uma CPU pode continuar outras tarefas enquanto uma interface de disco escreve dados no disco). Quando uma NIC termina a transmissão de um pacote, ela usa o mecanismo de interrupção do computador para informar a CPU.

Semelhantemente, uma NIC pode receber um pacote sem exigir o uso da CPU. Para receber um pacote, a CPU aloca espaço de buffer em memória e então instrui a NIC para ler o próximo pacote recebido no buffer. A NIC espera até que um quadro cruze a rede, faz uma cópia do quadro, verifica o checksum do quadro e o endereço de destino. Se o endereço de destino bate com o endereço do computador, o endereço de broadcast ou um endereço de multicast específico, a NIC armazena uma cópia do quadro na memória e interrompe a CPU. De outra forma, a NIC descarta o quadro e espera por outro. Desse modo, a NIC somente interrompe a CPU quando for recebido um quadro que o computador precise tratar. Para resumir:

A maioria das redes de computadores transfere dados através de um meio usando uma taxa fixa, freqüentemente mais rápida do que a velocidade em que os computadores podem processar bits individuais. Para acomodar o desencontro de velocidades, cada computador acoplado a uma rede contém hardware de propósito específico, conhecido como placa de interface de rede (NIC). A NIC funciona como um dispositivo de E/S: é construída para uma tecnologia de rede específica e trata dos detalhes de transmissão ou recepção de quadro sem exigir que a CPU processe cada bit.



Figura 10.2 A parte de trás de um computador com uma NIC instalada em um dos sockets. Um cabo liga o conector exposto à rede.

² NICs mais baratos não suportam transferências DMA; eles dependem então do CPU.

10.4 A conexão entre uma NIC e uma rede

O tipo de conexão usada entre uma NIC e uma rede depende da tecnologia da rede. Em algumas tecnologias, a NIC contém a maioria do hardware necessário e se liga diretamente ao meio de rede usando um cabo único ou uma fibra óptica. Em muitas outras tecnologias, a NIC não contém todos os circuitos eletrônicos necessários para ligação direta com a rede. Em vez disso, o cabo de uma NIC se liga a um componente eletrônico adicional que então se liga à rede.

Surpreendentemente, os detalhes exatos da conexão entre uma NIC e uma rede não são determinados pela tecnologia – uma tecnologia de rede pode suportar múltiplos esquemas de cabeamento. Para entender o conceito, será examinada uma tecnologia única que evoluiu através de três esquemas de cabeamento: a Ethernet. Embora a tecnologia subjacente seja sempre a mesma, os esquemas de cabeamento diferem drasticamente.

10.5 Cabeamento original espesso de Ethernet

O esquema de cabeamento original da Ethernet era informalmente chamado de *Ethernet de fio espesso* ou *Thicknet*, porque o meio de comunicação consiste em um cabo coaxial grande; o termo formal para o esquema é *10Base5*³. O hardware usado com Thicknet era dividido em duas partes importantes. A NIC continha os circuitos que tratavam dos aspectos digitais da comunicação, inclusive a detecção de erro e o reconhecimento de endereços. Por exemplo, a NIC gerava a CRC em cada quadro na transmissão e verificava a CRC em um quadro na recepção. A NIC verificava também o endereço de destino de um quadro e o passava para a CPU somente se ele fosse destinado ao computador. Finalmente, a NIC tratava de toda comunicação com o sistema do computador (ou seja, a NIC usava o barramento para transferir dados de/para a memória e o mecanismo de interrupção para informar a CPU de que uma operação fora concluída).

Uma NIC usada com Thicknet não incluia hardware analógico e, portanto, não tratava de sinais analógicos. Por exemplo, a NIC não detectava uma portadora, não convertia bits em voltagens apropriadas para transmissão e não convertia os sinais que estavam sendo recebidos para bits. Em vez disso, o hardware analógico que tratava de tais tarefas era colocado em um dispositivo eletrônico separado, chamado *transceiver*. Um transceiver era exigido para cada computador. Fisicamente, ele estava acoplado diretamente ao cabo de Ethernet, e um cabo separado o conectava à NIC em um computador. Deste modo, um transceiver normalmente era remoto a um computador. Por exemplo, em um edifício comercial, os transceivers poderiam estar acoplados a uma Ethernet no teto de um corredor.

O cabo conectando uma NIC a um transceiver é conhecido como um cabo de *Interface da Unidade de Anexo (Attachment Unit Interface, AUI)*, e os conectores na NIC e no transceiver são conhecidos como conectores de AUI. A Figura 10.3 mostra como um cabo de AUI conecta um computador a um transceiver.

Um cabo de AUI contém muitos fios. Claro, são necessários dois fios para transportar dados em transmissão da NIC até o transceiver e os dados em recepção do transceiver até a NIC. Além disso, um cabo de AUI contém fios separados que permitem à NIC controlar o transceiver e fios que transportam energia elétrica para ele.

A Figura 10.3 mostra outro detalhe do cabeamento de rede exigido por muitas tecnologias – a terminação de cabo. Cada extremidade do cabo coaxial que forma uma Ethernet deve ter um dispositivo de terminação, pequeno e barato, instalado. Um *terminador* consiste em um resistor que conecta o fio central no cabo à proteção. Essencialmente, quando um sinal elétrico alcança o terminador, ele é descartado. Curiosamente, a terminação é essencial para a correta operação de uma rede porque uma extremidade não-terminada de um cabo reflete sinais elétricos como um espelho reflete a luz. Se uma estação tenta enviar um sinal através de um cabo sem terminador, o sinal refle-

³ Ethernets modernas usam um mecanismo alternativo.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

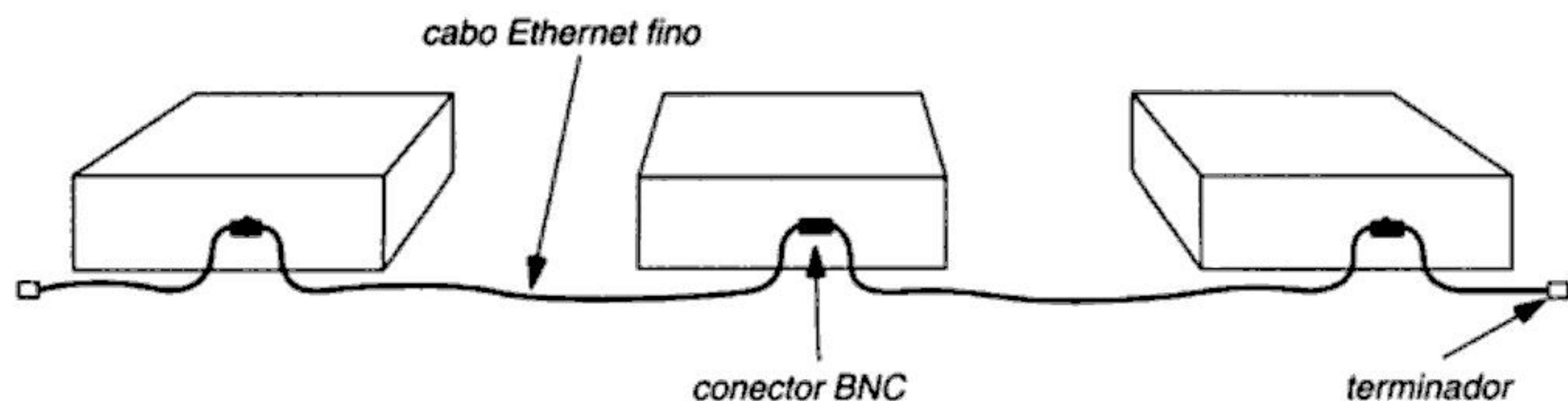


Figura 10.5 Três computadores conectados em uma Ethernet de fio fino. O meio é um cabo flexível que vai da NIC em um computador diretamente à NIC em outro computador.

Embora o cabeamento para uma Ethernet fina pareça ser completamente diferente do cabeamento para uma Ethernet espessa, os dois esquemas compartilham várias propriedades importantes. O cabo espesso e o fino são coaxiais, significando que eles protegem os sinais de interferência externa. Eles exigem terminação e usam a topologia de barramento. Mais importante, já que os dois sistemas de cabeamento têm características elétricas semelhantes (ou seja, resistência e capacidade), os sinais se propagam ao longo dos cabos da mesma maneira. Pode-se resumir:

O esquema de cabeamento de Thinnet para Ethernet usa um cabo coaxial flexível que se acopla diretamente a cada computador sem um transceiver separado. Embora sejam fisicamente diferentes, os cabos de Ethernet finos e espessos têm características elétricas semelhantes.

10.8 Ethernet de par trançado

Uma terceira geração de cabeamento de Ethernet mostra como os esquemas de cabeamento podem levar a topologias físicas inesperadas. Essa geração difere drasticamente da Ethernet fina e espessa porque não utiliza um cabo coaxial. Seguidamente chamado de Ethernet de par trançado, ou simplesmente de *Ethernet TP*, a primeira versão era formalmente conhecida como *10Base-T*. O cabo para a *Ethernet Rápida* é conhecido como *100Base-T*, e o cabeamento para a *Ethernet Gigabit* é conhecido como *1000Base-T*. Em vez de um meio compartilhado, a nova tecnologia estende a idéia usada em multiplexação de conexões: um dispositivo eletrônico serve como centro da rede. O dispositivo eletrônico básico é conhecido como *Ethernet Hub*⁴.

Como os outros esquemas de cabeamento, a Ethernet de par traçado exige que cada computador tenha uma placa de interface de rede e uma conexão direta da NIC até a rede. A conexão usa cabeamento de par trançado com conectores *RJ-45*, versões maiores dos conectores modulares usados com telefones. O conector em uma extremidade de um par trançado se conecta à interface de rede em um computador, e o conector na outra extremidade se conecta ao hub. Desse modo, cada computador tem uma conexão dedicada com o hub; não existe nenhum cabo coaxial. A Figura 10.6 mostra o cabeamento *10Base-T*.

Os componentes eletrônicos em um hub simulam um cabo físico, fazendo o sistema inteiro operar como uma Ethernet convencional. Por exemplo, um computador acoplado a um hub deve ter um endereço físico de Ethernet; cada computador deve usar a CSMA/CD para acessar a rede e o formato padrão de quadro Ethernet. O software não distingue entre Ethernet espessa, Ethernet fina e Ethernet de par trançado – a interface de rede trata dos detalhes e esconde quaisquer diferenças.

⁴ O próximo capítulo considera uma alternativa conhecida como *switch*.

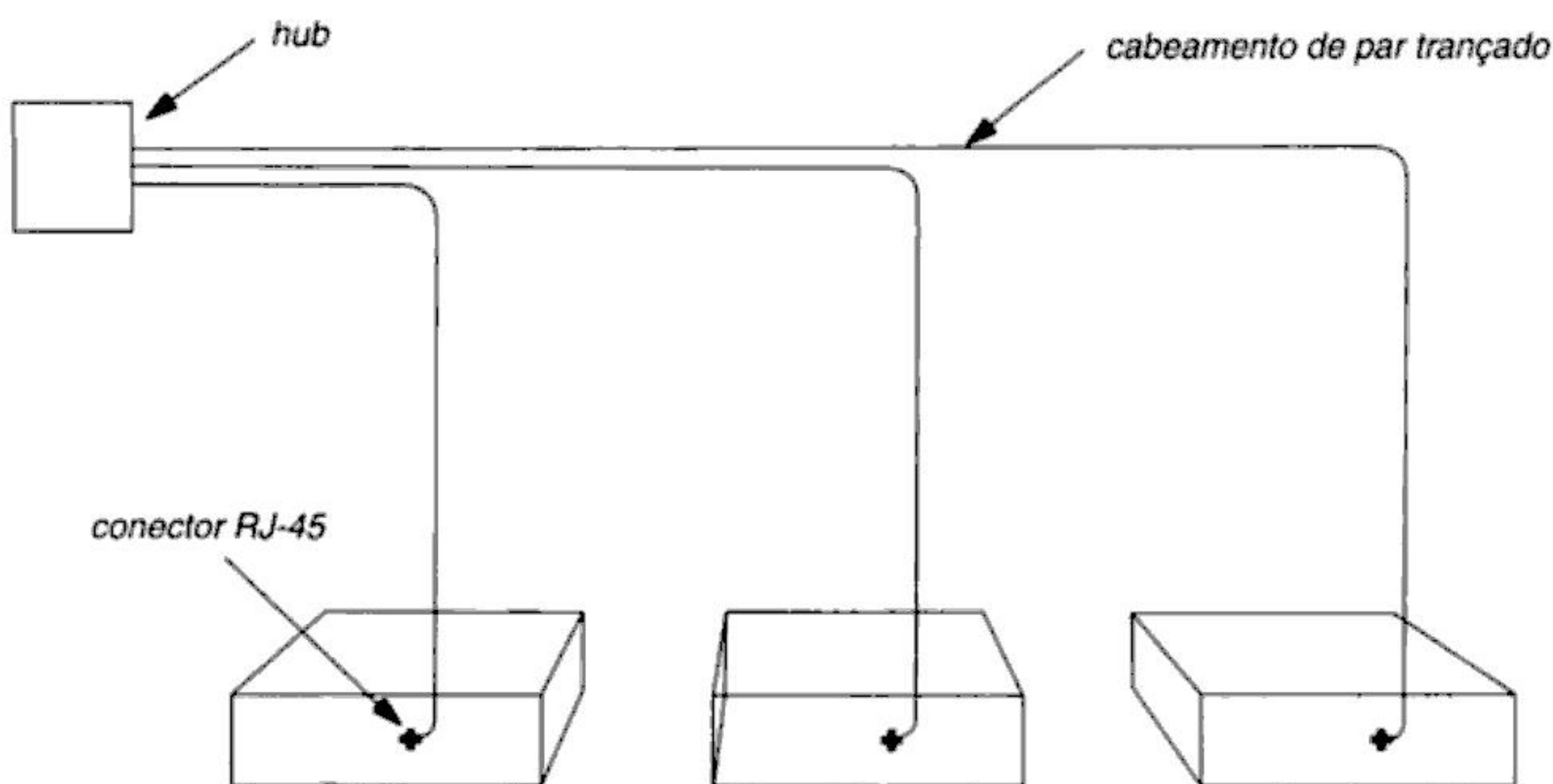


Figura 10.6 Três computadores conectados a um hub de Ethernet usando cabeamento de par trançado. Cada computador tem uma conexão dedicada.

Embora todos os hubs possam acomodar múltiplos computadores, eles vêm em uma variedade de tamanhos. Um hub pequeno típico tem quatro ou cinco *portas*, cada uma aceitando uma conexão. Deste modo, um hub é suficiente para conectar todos os computadores em grupos pequenos (por exemplo, em um único departamento). Hubs maiores podem acomodar centenas de conexões.

O esquema de cabeamento para Ethernet de par trançado usa um dispositivo eletrônico conhecido como hub em lugar de um cabo compartilhado. A conexão entre um computador e o hub emprega cabeamento de par trançado.

10.9 Vantagens e desvantagens de esquemas de cabeamento

Cada um dos três esquemas de cabeamento tem vantagens e desvantagens. O cabeamento que usa um transceiver separado para cada conexão permite que um computador seja trocado sem interromper a rede. Quando um cabo de transceiver for desconectado, um transceiver perde energia, mas outros continuam a operar. Ter transceivers separados apresenta desvantagens. Eles são freqüentemente posicionados em locais remotos de difícil acesso (por exemplo, um teto de corredor em um edifício comercial). Se um transceiver falha, encontrar, testar ou substituir o mesmo pode ser tedioso. Em contraste, embora não tenha a desvantagem de transceivers remotos, um esquema em que o meio compartilhado está diretamente conectado a cada computador é suscetível a desconexão – desconectar o cabo principal deixaria os segmentos sem terminação, rompendo a rede inteira. Além disso, tal rompimento é provável porque, diferentemente de conectores de AUI, nenhuma ferramenta é necessária para desconectar conectores BNC usados com Thinnet. O cabeamento de hub torna a rede menos exposta à desconexão acidental, já que cada par trançado afeta somente uma máquina. Desse modo, se um fio único for accidentalmente cortado, somente uma máquina será desconectada do hub.

Apesar das vantagens e desvantagens mencionadas acima, um fator parece dominar a escolha de tecnologia de cabeamento: o custo. A Ethernet fina se tornou popular porque custa menos por conexão do que a Ethernet espessa original. O cabeamento 10Base-T é agora popular porque custa menos por conexão do que Ethernet fina. Claro, muitas afirmações são generalizações – os custos reais nem sempre são fáceis de comparar. O custo total depende do número de computadores, da distância entre eles, da colocação física de paredes e dutos, do custo do hardware de interface e ca-

beamento, do custo para diagnosticar e consertar problemas e da freqüência com que computadores novos são acrescentados ou computadores existentes repositionados. Como a maioria das organizações usa um esquema de cabeamento único para ligar computadores a uma determinada rede, as interfaces devem estar disponíveis a um custo razoável para que todas as marcas de computadores possam ser conectadas. Assim, nenhum esquema de cabeamento único é melhor para todas as situações. Mais importante, como todos os esquemas de cabeamento usam o mesmo padrão para formato de quadro e acesso à rede, é possível construir dispositivos que passem quadros de um tipo de cabo para outro. Por exemplo, é possível conectar uma rede Ethernet Thicknet à uma rede de Ethernet Thinnet e permitir que quadros sejam transmitidos entre elas.

Para visualizar algumas das diferenças em custo, imagine um conjunto inteiro de escritórios com um ou mais computadores em cada. A Figura 10.7 mostra como os três esquemas de cabeamento de Ethernet poderiam aparecer em um conjunto de oito escritórios separados por um corredor em comum.

10.10 O paradoxo da topologia

Um leitor atento terá notado uma aparente contradição na descrição da tecnologia de Ethernet. O Capítulo 8 afirma que a Ethernet usa uma topologia de barramento. Neste capítulo, tal afirmação nem sempre parece ser verdadeira. O barramento é aparente em cabeamento espesso ou fino porque o barramento compartilhado é o cabo coaxial. Porém, uma Ethernet usando par trançado como cabeamento não se assemelha a um barramento. De fato, segundo as definições no Capítulo 8, o cabeamento de par trançado forma uma topologia de estrela, tendo o hub como centro.

A Ethernet é uma topologia de barramento, ou a topologia depende do cabeamento? As duas respostas estão corretas! Obviamente, uma Ethernet de par trançado forma uma estrela clássica, em que cada computador tem uma conexão dedicada para o hub central. Apesar de sua aparência, porém, a Ethernet de par trançado funciona como um barramento. Todos os computadores compartilham um meio de comunicação único. Os computadores devem competir no acesso ao meio, e no máximo um computador pode transmitir em um determinado momento. Como na Ethernet convencional, as interfaces de rede em todos os computadores recebem uma cópia de cada pacote transmitido, e a interface de rede é responsável por filtrar pacotes exatamente do mesmo modo que a interface filtra pacotes que chegam de uma Ethernet de fio espesso ou de fio fino. Como resultado, quando um computador envia um quadro para o endereço broadcast, todos os outros computadores recebem uma cópia.

Para resolver o paradoxo aparente e entender as tecnologias de rede, deve-se distinguir entre topologias físicas e lógicas. Fisicamente, a Ethernet de par trançado usa uma topologia de estrela. Logicamente, a Ethernet de par trançado funciona como um barramento. Desse modo, uma Ethernet de par trançado é freqüentemente chamada de *barramento em forma de estrela*⁵. Para resumir:

Uma dada tecnologia de rede pode usar uma variedade de esquemas de cabeamento. A tecnologia determina a topologia lógica, e o esquema de cabeamento determina a topologia física. É possível diferir a topologia física da topologia lógica.

10.11 Placas de interface de rede e esquemas de cabeamento

Como contém circuitos para tratar dos detalhes elétricos da comunicação, uma interface de rede deve suportar um esquema de cabeamento assim como uma tecnologia de rede. Por exemplo, uma interface para Ethernet de par trançado deve ter um conector RJ-45 e gerar sinais de acordo com

⁵ Informalmente, um par trançado Ethernet é chamado “barramento na caixa” (*bus in a box*) ou “rede na caixa” (*network in a box*).

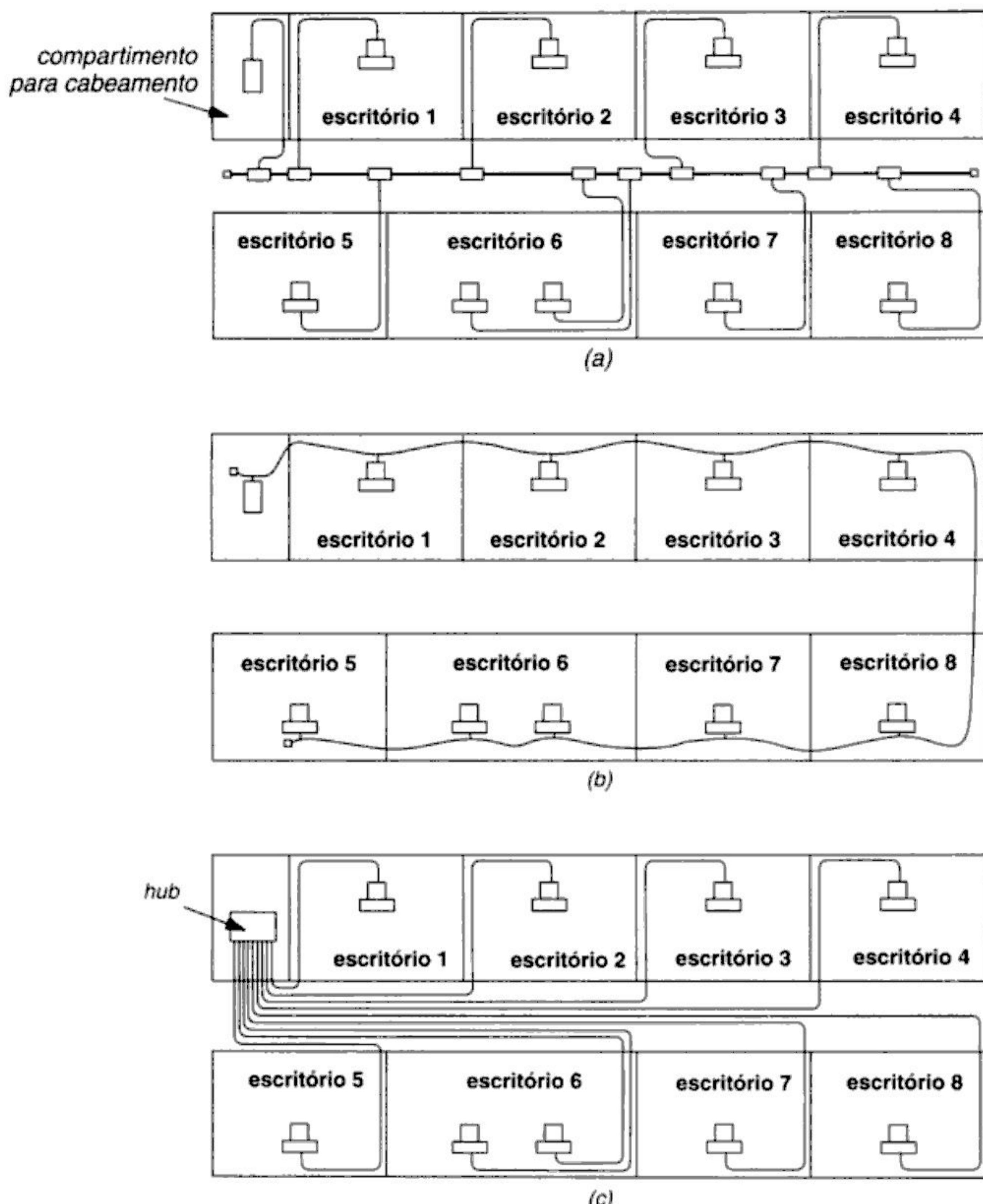


Figura 10.7 Ilustração de computadores em oito escritórios com cabeamento (a) espesso, (b) fino e (c) de par trançado para Ethernet. Os fios podem correr acima do teto ou debaixo de um piso falso. Um armário para cabeamento pode conter um hub ou equipamento usado para monitoração de rede, controle ou depuração.

a especificação 10Base-T. Entretanto, uma interface usada com Ethernet fina deve ter um conector BNC e gerar sinais apropriados para Thinnet. Para possibilitar a mudança de esquemas de cabeamento sem mudar o hardware de interface, muitas interfaces de rede suportam múltiplos esquemas de cabeamento. Por exemplo, freqüentemente uma única NIC de Ethernet tem três conectores, como mostra a Figura 10.8.

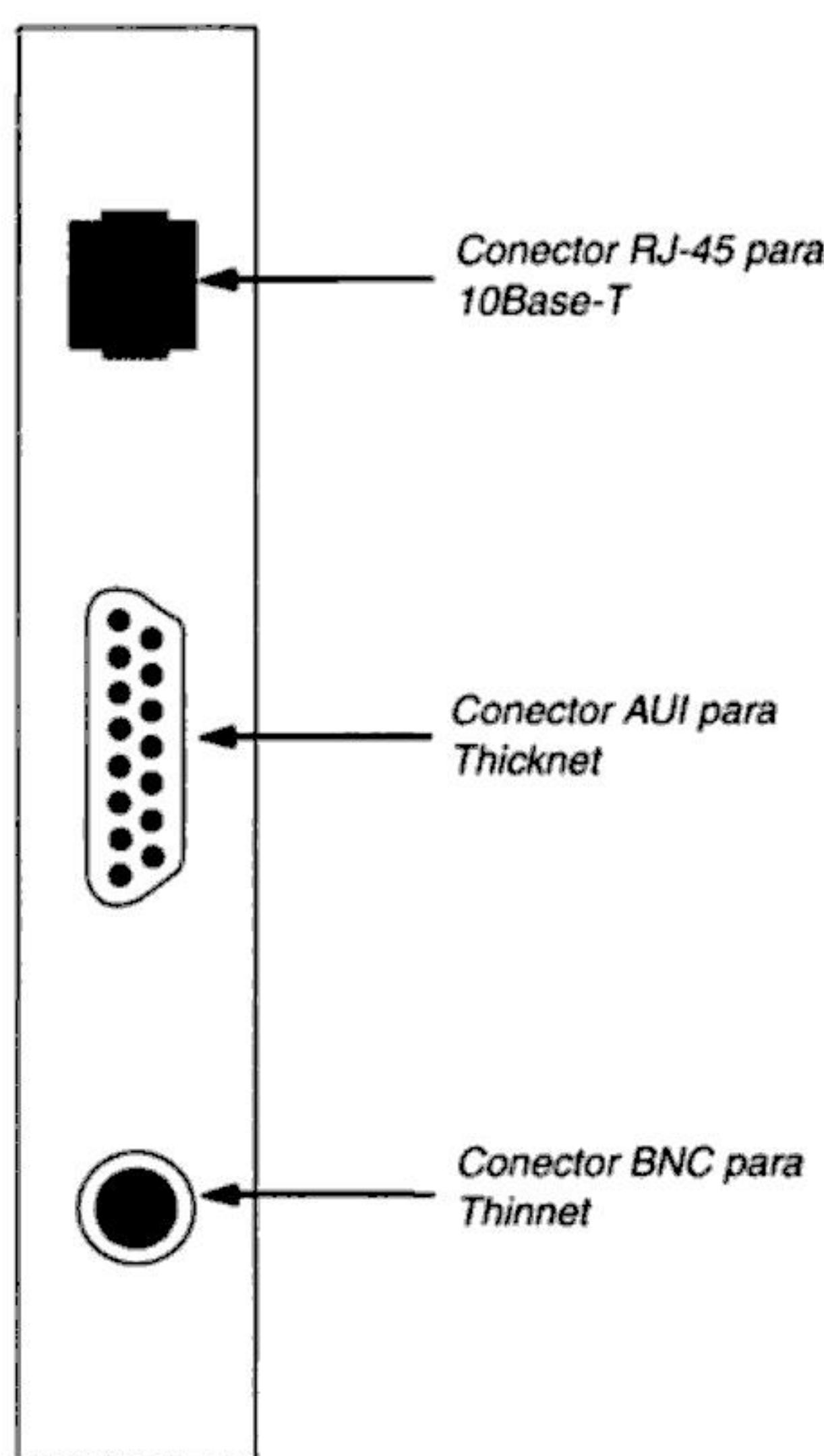


Figura 10.8 Ilustração da parte exposta da placa de interface de Ethernet quando as placas estão instaladas em um computador. A interface pode ser usada com um dos três esquemas de cabeamento básico. Cada esquema de cabeamento usa um conector de estilo diferente.

Embora múltiplos conectores fiquem permanentemente conectados, uma determinada interface pode usar apenas um esquema de cabeamento de cada vez. O software no computador deve ativar um dos conectores; os demais não são usados. A vantagem principal de ter múltiplos esquemas de cabeamento suportados por uma única NIC é a flexibilidade – um site pode escolher um esquema de cabeamento ou mudar para um cabeamento diferente sem substituir o hardware de interface. Mais importante, como o endereço físico de um computador é atribuído à NIC, ele permanece constante ao passar para um novo esquema de cabeamento.

10.12 Interfaces de rede 10/100 e auto-negociação

Foi dito que a versão original 10Base-T da Ethernet de par trançado operava a 10 Mbps, que a Ethernet 100Base-T operava a 100 Mbps e que a Ethernet 1000Base-T operava a Gbps. Os padrões para os três esquemas de par trançado foram designados para usar conectores RJ-45. Mais importante, a tecnologia 100Base-T é projetada para ser compatível para trás (backward), e para permitir aos participantes que a velocidade seja negociada quando uma conexão física for estabelecida. Durante o processo, conhecido como *autonegotiação*⁶, os dois dispositivos informam um ao outro a velocidade que

⁶ Antes da autonegotiação ser padronizada, alguns vendedores produziam soluções para proprietários e usavam o nome *autosensing*.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

trançado usa uma conexão separada para cada computador até um dispositivo eletrônico chamado *hub*. O hub simula um cabo compartilhado.

Um princípio importante surge dos exemplos de esquemas de cabeamento: a topologia física da rede pode diferir da topologia lógica. Por exemplo, um esquema de cabeamento que usa um hub é, fisicamente, uma estrela. Porém, os circuitos eletrônicos no hub podem simular uma topologia lógica diferente (por exemplo, um barramento ou um anel). Tal rede é conhecida como *barramento em forma de estrela ou anel em forma de estrela*.

Para acomodar múltiplos esquemas de cabeamento, freqüentemente uma interface de rede vem com múltiplos conectores. Somente um conector pode estar ativo de cada vez – o software no computador deve escolher qual conector usar.

Para estudos futuros

Vários grupos produziram padrões de cabos. O *American National Standards Institute (ANSI)*, a *Telecommunications Industry Association (TIA)* e a *Electronic Industries Association (EIA)* cooperaram para definir padrões de cabos. Outras companhias e organizações fornecem certificados e testes para cabos e conectores.

Exercícios

- 10.1 Que tipo de cabeamento sua organização usa? Descubra por que a escolha foi feita.
- 10.2 De que tamanho é uma NIC? Veja se você pode achar uma NIC de Ethernet fabricada vários anos atrás. Compare-a com uma NIC que caiba em uma placa PCMCIA.
- 10.3 Em uma Ethernet de fio fino, dois cabos são conectados a um conector atrás de cada computador. O que acontece se um usuário desconecta o conector BNC do computador? O que acontece se um usuário desconecta um dos dois cabos?
- 10.4 Leia sobre produtos comerciais de hub. Quantos computadores podem ser conectados a um hub pequeno? E a um hub grande?

Sumário do Capítulo

- 11.1** Introdução 163
- 11.2** Limitação de Distância e Projeto de LANs 163
- 11.3** Extensões de Fibra Óptica 164
- 11.4** Repetidores 165
- 11.5** Bridges 167
- 11.6** Filtragem de Quadros 168
- 11.7** Comportamento Inicial e Estado Estável de Redes com Bridges 168
- 11.8** Planejamento de uma Rede com Bridges 169
- 11.9** Ligação com Bridge entre Edifícios 170
- 11.10** Uso de Bridges Através de Distâncias Mais Longas 170
- 11.11** Um Ciclo de Bridges 172
- 11.12** Distributed Spanning Tree (Árvore de Extensão Distribuída) 173
- 11.13** Comutação (Switching) 173
- 11.14** Combinando Switches e Hubs 174
- 11.15** Ligação por Bridges e Comutação com Outras Tecnologias 175
- 11.16** Resumo 175

Estendendo LANs: Modems de Fibra, Repetidores, Bridges e Switches

11.1 Introdução

Os capítulos anteriores descrevem tecnologias básicas de LANs e esquemas de cabeamento afins. Cada tecnologia de LAN é projetada para uma combinação específica de velocidade, distância e custo. O projetista especifica uma distância máxima que a LAN pode alcançar, com LANs típicas projetadas para alcançar algumas centenas de metros. Como resultado, a tecnologia de LAN funciona melhor para conectar computadores dentro de um único edifício.

Infelizmente, as pessoas que interagem eletronicamente nem sempre ocupam escritórios localizados dentro de poucas centenas de metros. Este capítulo discute mecanismos que podem estender uma LAN por distâncias mais longas e usa modems de fibra, repetidores e bridges para ilustrar algumas das possibilidades.

11.2 Limitação de distância e projeto de LANs

A limitação de distância é uma parte fundamental dos projetos de LANs. Quando estão projetando uma tecnologia de rede, os engenheiros escolhem uma combinação de capacidade, atraso máximo e distância que pode ser alcançada a um dado custo. Para ajudar a diminuir os custos, as tecnologias de LANs normalmente usam um meio de comunicação compartilhado, como um barramento compartilhado ou um anel. Como consequência do uso de um meio compartilhado, um projeto de LAN deve incluir um mecanismo que garanta a cada estação acesso justo ao meio compartilhado. Por exemplo, algumas tecnologias de barramento usam a CSMA/CD, enquanto as tecnologias de anel usam passagem de token.

A necessidade de um mecanismo de acesso justo é uma das motivações principais para limitar o comprimento de uma LAN. Os dois mecanismos de acesso mais populares, CSMA/CD e passagem de token, levam um tempo proporcional ao tamanho da rede. Para assegurar que os atrasos não se tornarão significativos, uma tecnologia de LAN é projetada para funcionar com um valor fixo para comprimento máximo de cabo.

Surge outra limitação porque o hardware é projetado para emitir uma quantia fixa de energia elétrica. Infelizmente, como um sinal elétrico se torna gradualmente mais fraco enquanto viaja ao longo de um fio de cobre, ele não pode alcançar uma distância arbitrariamente longa. Para assegu-

rar que todas as estações acopladas a uma LAN recebam um sinal suficientemente forte, os projetistas calculam o comprimento máximo de fio permitido. Deste modo,

Uma especificação de comprimento máximo é uma parte fundamental da tecnologia de LANs. O hardware de LAN é projetado para um cabo de comprimento máximo fixo e não trabalhará corretamente através de fios que excedam a esse limite.

11.3 Extensões de fibra óptica

Os engenheiros desenvolveram uma variedade de formas para estender a conectividade das LANs. Em geral, os mecanismos não aumentam a força dos sinais elétricos gerados por hardware de interface, nem meramente acrescentam fio para estender os cabos além do limite máximo. Em vez disso, a maioria dos mecanismos de extensão usa hardware de interface padrão e insere componentes de hardware adicional que podem encaminhar sinais através de distâncias mais longas.

O mecanismo de extensão de LANs mais simples insere fibras ópticas e um par de *modems de fibra* entre um computador e um transceiver. Como as fibras têm baixo atraso e banda alta, esse mecanismo pode permitir que um computador se conecte a um transceiver que, por sua vez, está acoplado a uma rede remota. O computador envia sinais-padrão, e o transceiver os recebe. Desse modo, a extensão pode funcionar com hardware de interface de rede padrão. A Figura 11.1 ilustra modems de fibra usados para estender uma conexão de Ethernet.

Como mostra a figura, um modem de fibra reside em cada extremidade da conexão e uma fibra óptica os conecta. O computador usa uma interface de rede que gera sinais convencionais para se comunicar com a rede (por exemplo, um hub de Ethernet) e envia os sinais para o modem de fibra local. De forma semelhante, o modem de fibra remoto gera sinais-padrão de AUI e os envia para o hub.

Cada um dos modems de fibra contém um hardware para executar duas tarefas: os circuitos eletrônicos no modem de fibra convertem sinais de AUI em representação digital, e o hardware de controlador óptico traduz a representação digital em pulsos de luz que viajam ao longo da fibra. Claro, os circuitos devem fornecer comunicação em ambas as direções para permitir que o computador envie e receba quadros¹. Então, os circuitos no modem da fibra próximo ao computador devem aceitar dados digitais que chegam através da fibra e convertê-los em sinais, que são enviados ao computador, como também aceitar sinais do computador e convertê-los em dados digitais destinados ao hub.

A principal vantagem dos modems de fibra surge da sua capacidade de fornecer uma conexão para uma LAN remota sem mudar a LAN original ou o computador. Como os atrasos através de fibras

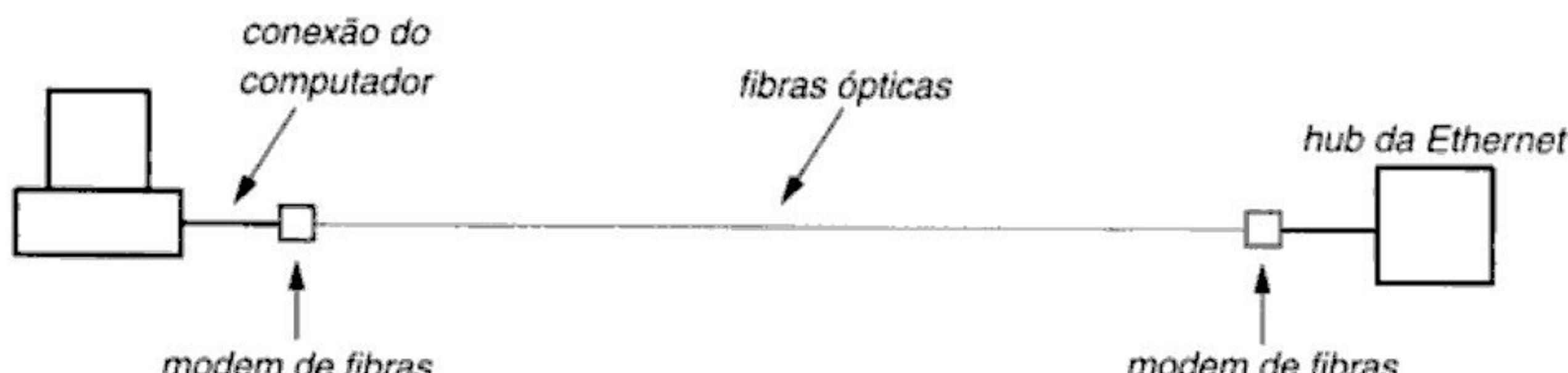


Figura 11.1 Fibras ópticas e modems de fibras usados para fornecer uma conexão entre um computador e uma Ethernet distante. O computador e o hub da Ethernet usam sinais convencionais.

¹ Muitas implementações usam um par de fibras para permitir a transmissão simultânea em ambas as direções.

são baixos e a largura de banda é alta, o mecanismo irá operar corretamente a distâncias de vários quilômetros. O uso mais comum envolve conectar um computador em um edifício a uma LAN em outro edifício.

Para resumir:

Um par de modems de fibra e fibras ópticas pode ser usado para fornecer uma conexão entre um computador e uma LAN remota. O mecanismo é inserido entre a interface de rede em um computador e um hub remoto.

11.4 Repetidores

Recorde que uma limitação de distância em LANs surge porque um sinal elétrico se torna mais fraco ao viajar ao longo de um fio. Para superar tal limitação, algumas tecnologias de LAN permitem que duas LANs sejam unidas por meio de um dispositivo conhecido como *repetidor*. Um repetidor é normalmente um dispositivo eletrônico analógico que continuamente monitora sinais elétricos em cada LAN. Quando percebe um sinal em uma LAN, ele transmite uma cópia ampliada na outra. A Figura 11.2 ilustra um repetidor usado com a Ethernet.

Como mostra a figura, um repetidor conecta dois cabos de Ethernet conhecidos como *segmentos*, cada um com sua terminação habitual. O repetidor não entende o formato de quadro, nem possui endereço físico. Em vez disso, ele se acopla diretamente aos cabos de Ethernet e envia cópias de sinais elétricos de um para o outro, sem esperar por um quadro completo.

O tamanho máximo do cabeamento espesso original da Ethernet era de 500 metros². A Figura 11.2 mostra que um repetidor pode dobrar o comprimento efetivo de uma Ethernet para 1.000 metros conectando dois segmentos de tamanho máximo. Um par de repetidores pode ser usado para conectar três segmentos de Ethernet, compondo uma rede de 1.500 metros de comprimento. Como os repetidores propagam todos os sinais entre dois segmentos, um computador conectado a um segmento pode se comunicar com um computador conectado a outro. Na verdade, quando estão usando repetidores, os computadores de origem e destino não podem determinar se estão conectados ao mesmo segmento ou em segmentos diferentes. Para resumir:

Um repetidor é um dispositivo de hardware usado para estender uma LAN. O repetidor, que conecta dois segmentos de cabo, amplifica e envia todos os sinais elétricos que acontecem em um segmento para o outro segmento. Qualquer par de computadores na LAN estendida pode se comunicar; os computadores não sabem se um repetidor os separa.

É possível construir uma Ethernet arbitrariamente longa usando repetidores para conectar muitos segmentos? A resposta é não. Embora tal organização garanta força suficiente, cada repetidor e ca-

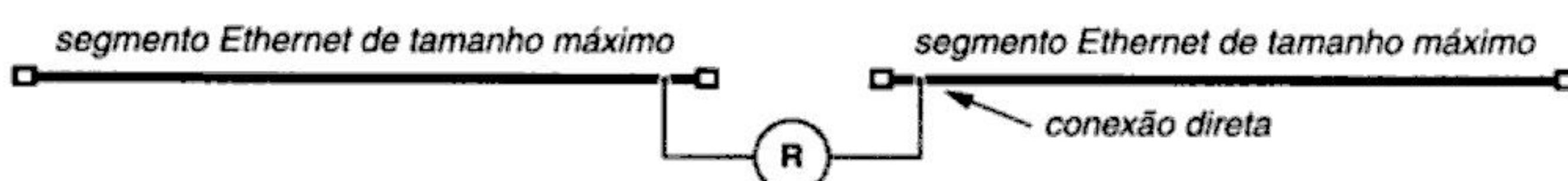


Figura 11.2 Um repetidor *R* conectando duas Ethernets. O repetidor é conectado diretamente ao cabo.

² O Thinnet e o cabeamento 10base-T têm limites menores.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

11.9 Ligação com bridge entre edifícios

Descrevemos algumas razões para usar bridges em um ambiente local. Como os repetidores, as bridges também podem ser usadas para alcançar distâncias mais longas. Por exemplo, uma corporação pode precisar de uma rede que permita que os computadores em um edifício se comuniquem com computadores em outros. Se os dois edifícios forem separados por uma distância significativa ou se forem grandes, uma única LAN não bastará. Mais importante, usar pares de modems de fibra para ligar todos os computadores de uma única LAN pode resultar em um custo alto ou desempenho abaixo do desejado.

Como uma bridge se conecta a uma LAN do mesmo modo que um computador, a forma mais simples de estender uma LAN com bridge por uma longa distância é empregar uma técnica já descrita neste capítulo. Uma fibra óptica e um par de modems de fibra são usados para estender uma das conexões entre uma bridge e um segmento de LAN, permitindo que o segmento possa ser localizado remotamente em relação à bridge. A Figura 11.6 mostra como podem ser usados modems de fibra para unir através da bridge dois segmentos de LAN em dois edifícios⁴.

O uso de uma bridge em tais situações tem três vantagens primárias. Primeiro, como exige somente uma conexão de fibra, a solução com bridge é mais barata do que usar uma conexão de fibra separada para cada computador individual. Segundo, como a conexão entre edifícios está acoplada à bridge, os computadores individuais podem ser acrescentados ou removidos dos segmentos sem ser necessário instalar ou mudar o cabeamento entre os edifícios. Terceiro, já que as bridges permitem a comunicação simultânea nos dois segmentos, usá-las em vez de um repetidor significa que a comunicação entre computadores em um prédio não tem impacto na comunicação entre os computadores em outro.

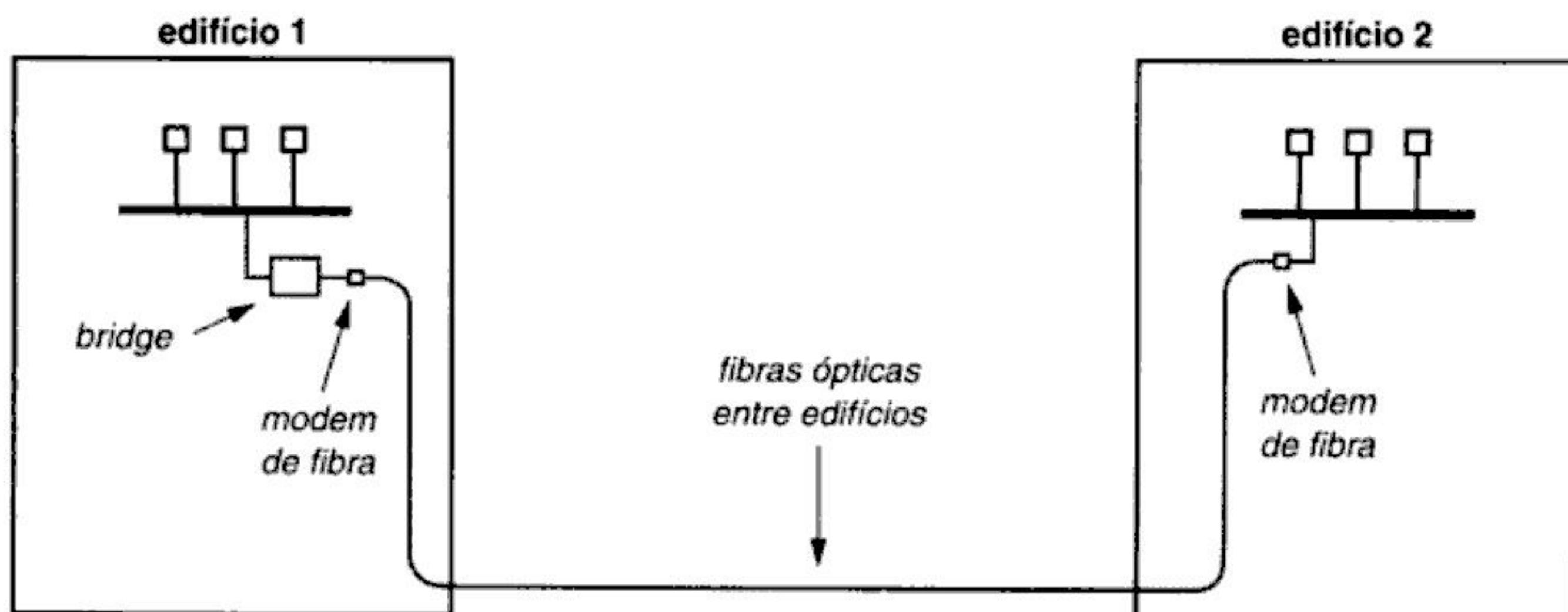


Figura 11.6 Uma bridge conectando segmentos de LAN em dois edifícios. Uma fibra óptica é usada para conectar a bridge a um segmento de LAN remoto.

11.10 Uso de bridges através de distâncias mais longas

Na maioria dos países, há leis que impedem uma organização de conectar sites com fibra óptica, a menos que a organização possua todas as propriedades entre os locais e que a fibra não precise

⁴ Talvez as bridges de longas distâncias usadas mais amplamente sejam aquelas implementadas por uma DSL ou cable-modem, como será descrito no próximo capítulo. Tais modems fornecem ligações com bridges entre a rede na propriedade de um assinante e a rede em uma ISP.

cruzar uma via pública. As organizações freqüentemente descobrem que a maioria da comunicação se dá entre computadores dentro de cada local, e que a comunicação entre sites é infreqüente. Uma LAN com bridges fornece uma solução geral para tais situações: a organização coloca um segmento de LAN em cada site e usa um par de bridges para conectar os segmentos.

Como uma rede com bridges pode alcançar longas distâncias? Dois métodos são populares. Cada um envolve uma conexão ponto a ponto de longa distância e hardware especial de bridge. O primeiro método usa uma linha serial alugada para conectar os sites, e o segundo usa um canal de satélite alugado. O uso de uma linha serial alugada é mais comum por ser mais barata. Porém, uma conexão de satélite é interessante por permitir a comunicação através de uma distância arbitrária. A Figura 11.7 mostra como uma bridge pode ser usada em uma conexão via satélite.

Como mostra a figura, o hardware usado com uma bridge de longa distância difere ligeiramente daquele usado em uma conexão local. Além de um segmento de LAN e a estação terrestre usada para comunicação via satélite, cada local tem o hardware de bridge. O hardware de bridge aprende os endereços de computadores locais no lugar e evita encaminhar quadros destinados a tais computadores.

A motivação para filtragem em ambos os locais deriva das restrições de largura de banda. Diferentemente da conexão de fibra óptica usada entre edifícios, LANs com bridge conectadas via circuitos alugados freqüentemente usam conexões com baixa largura de banda para reduzir os custos. Conseqüentemente, um canal de satélite típico usado para ligação com bridge opera com muito menos capacidade do que um segmento de LAN⁵. Como resultado, o canal de satélite não tem largura de banda suficiente para transportar todos os quadros de um segmento de LAN até o outro local para filtragem. Em vez disso, o hardware de bridge em cada extremidade do canal aprende os endereços dos computadores em seu local e não encaminha quadros, a menos que seja necessário.

Além da filtragem, o hardware de bridge usado com conexões de longa distância deve executar a *bufferização* pois os quadros podem chegar da rede local mais rapidamente do que podem ser enviados através do satélite. Bufferização significa armazenar uma cópia de cada quadro na memória até que ele possa ser enviado. Em essência, a bridge mantém uma fila de quadros aguardando transmissão. Depois da bridge receber um quadro da rede local e determinar se ele deve ser enca-

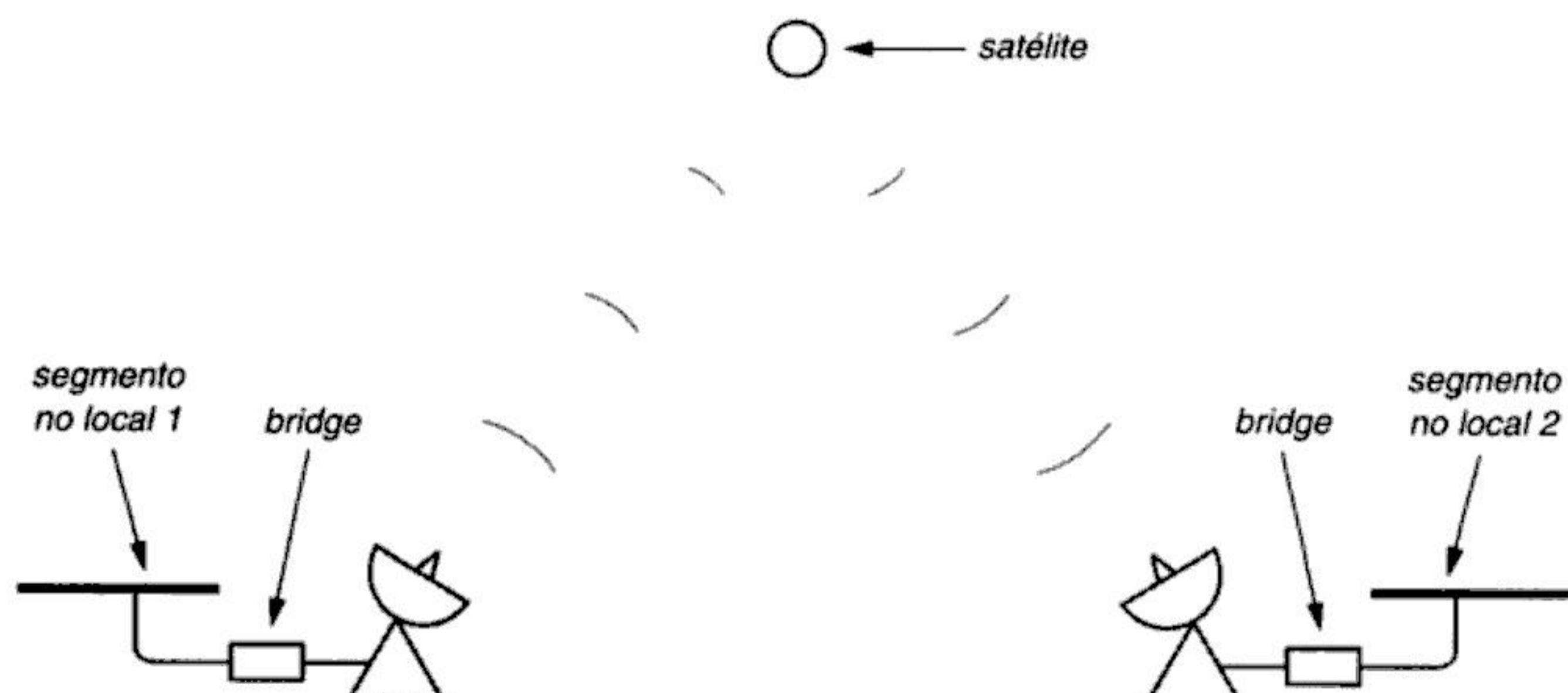


Figura 11.7 Uma bridge usando um canal de satélite alugado para conectar segmentos de LAN entre dois locais. Uma bridge via satélite pode alcançar distâncias arbitrárias.

⁵ Um canal de satélite operacional a 56 Kbps oferece menos de um por cento da capacidade de uma LAN típica; uma linha serial alugada que usa padrão telefônico T1 tem aproximadamente 15% da capacidade de uma LAN típica.

minhado para o outro local, ela acrescenta o quadro à fila. Se o transmissor de satélite está inativo, a bridge começa a enviar o quadro. Quando o hardware de satélite terminar o envio do quadro, ele automaticamente começa a enviar o próximo na lista.

Claro, a bufferização não resolve completamente o problema – se os quadros continuarem a chegar da LAN mais rapidamente do que o satélite pode enviá-los, a bridge ficará sem memória e começará a descartá-los. Porém, a maioria dos softwares de comunicação espera por uma resposta após enviar alguns quadros. Dessa forma, a memória da bridge é suficiente para guardar os quadros até que eles possam ser transmitidos.

11.11 Um ciclo de bridges

Como uma bridge envia e recebe quadros, uma rede com bridge pode alcançar muitos segmentos. Por exemplo, a Figura 11.8 mostra oito segmentos de LAN conectados por bridges.

Como mostra a figura, é necessária uma bridge para conectar cada segmento com o resto da rede ligada por bridges. Embora cada uma delas introduza um pequeno atraso, a rede encaminha corretamente um quadro de um computador em qualquer segmento para outro computador em qualquer outro segmento. Por exemplo, suponha que um computador em um segmento *g* envie um quadro para um computador em segmento *e*. A bridge B_6 encaminhará uma cópia da transmissão original para o segmento *d*, a B_3 encaminhará uma cópia para o segmento *b*, e assim por diante até que uma cópia alcance o segmento *E*.

O broadcast funciona em ambientes com bridges porque uma bridge sempre encaminha uma cópia de um quadro enviado para o endereço de broadcast. Por exemplo, se um computador no segmento *g* transmite um quadro, a bridge B_6 encaminha o broadcast para o segmento *d*. As bridges B_3 e B_7 receberão a cópia no segmento *d* e a encaminharão para os segmentos *b* e *h*. A cópia no segmento *b* alcançará B_1 e será, mais cedo ou mais tarde, propagada para todos os segmentos restantes.

Nem todas as bridges deveriam ter permissão para encaminhar quadros de broadcast, ou um ciclo de bridges causará problema. Para entender por quê, observe a Figura 11.9, que mostra quatro segmentos interconectados por quatro bridges. Considere o que acontece se um computador em um segmento enviar um quadro broadcast. A bridge B_1 encaminha uma cópia para o segmento *b*, enquanto B_2 encaminha uma cópia para o segmento *c*. Quando a B_4 recebe a cópia enviada pela B_2 , ela encaminha aquela cópia para o segmento *d*. Semelhantemente, quando a B_3 recebe a cópia enviada pela B_1 , a B_3 encaminha tal cópia para o segmento *d*. Deste modo, os computadores acoplados ao segmento *d* recebem múltiplas cópias. Mais importante, quando a cópia de B_4 viaja através do segmento *d, B_3 encaminha aquela cópia para o segmento *b*. Semelhantemente, quando a cópia de B_3 viaja através do segmento *d, B_4 a encaminha para o segmento *c*. A menos que alguma brid-**

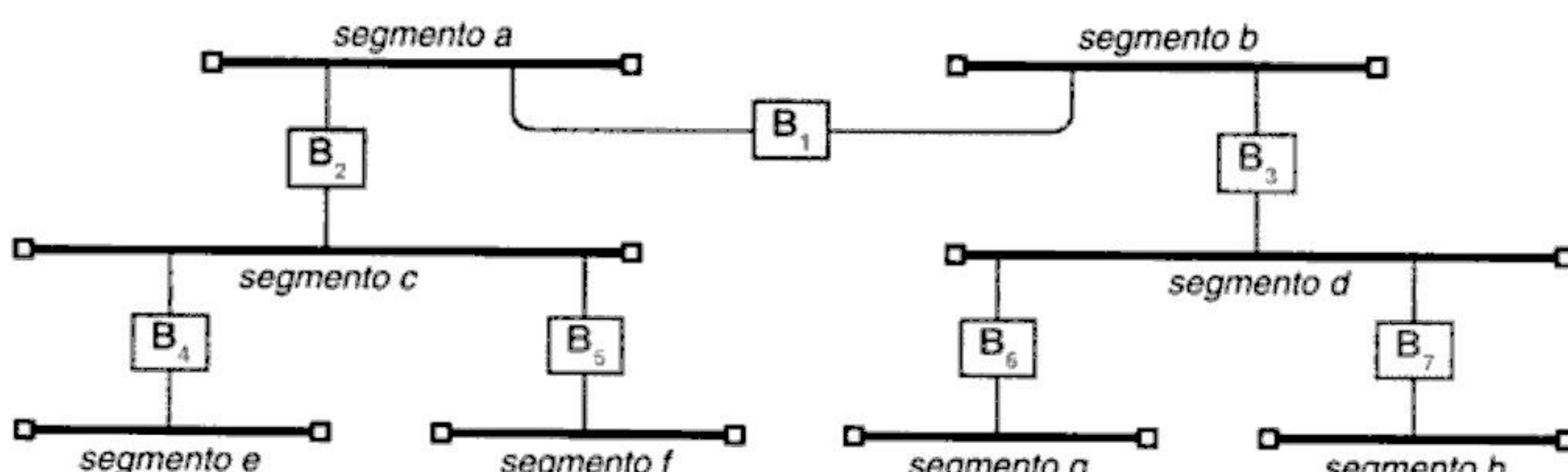


Figura 11.8 Uma rede com bridges que consiste em oito segmentos conectados por sete bridges. Os computadores podem estar acoplados a qualquer segmento.

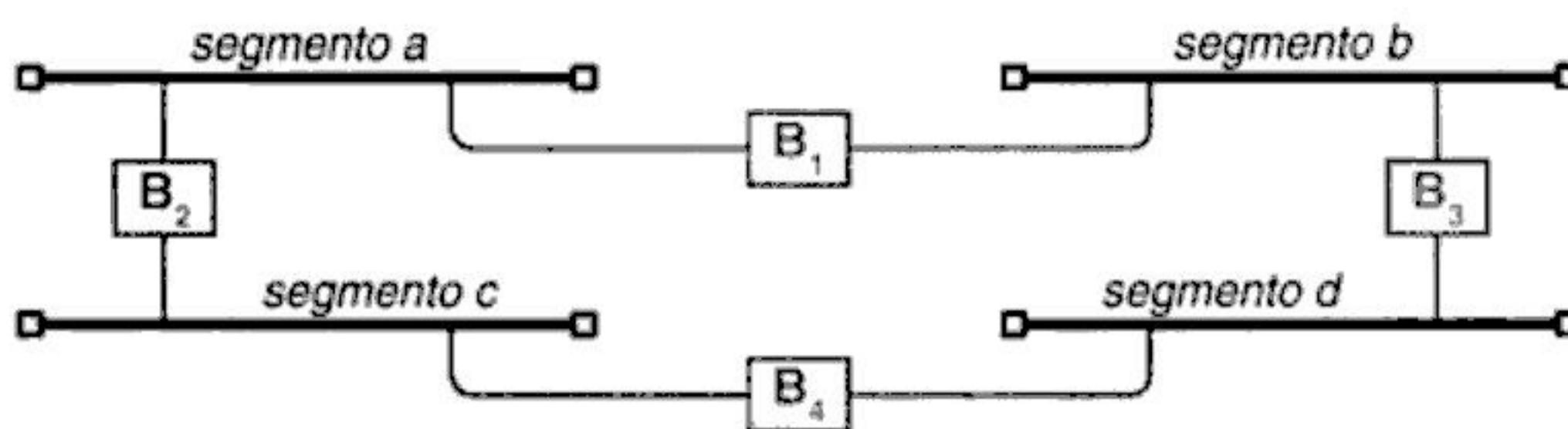


Figura 11.9 Um exemplo de bridges conectadas em um ciclo. Haverá problema se todas as bridges encaminharem quadros de broadcast.

ge seja impedida de encaminhar broadcasts, as cópias continuam a fluir em torno do ciclo para sempre, com computadores em todos os segmentos recebendo um número infinito de cópias.

11.12 Distributed spanning tree (árvore de extensão distribuída)

Para prevenir o problema de laços (loop) infinitos, uma rede unida por bridge não deve permitir que as condições abaixo aconteçam simultaneamente:

- Todas as bridges encaminham todos os quadros.
- A rede unida por bridges contém um ciclo de segmentos ligados por bridges.

Na prática, pode ser difícil prevenir que ciclos sejam accidentalmente introduzidos em uma grande rede unida por bridges que se espalha por uma organização. Além disso, as organizações às vezes escolhem colocar bridges extras em uma rede para torná-la mais imune a falhas. Para impedir que os laços aconteçam, algumas das bridges em uma rede unida por bridges deve concordar em não encaminhar quadros.

O esquema usado para prevenir laços em uma rede unida por bridges é interessante porque é automatizado. Um local pode conectar bridges em uma configuração arbitrária e permitir que elas operem sem configurar manualmente quais bridges encaminharão broadcasts – as bridges se configuram automaticamente.

Como uma bridge pode saber se deve encaminhar quadros? Quando uma bridge é inicializada, ela se comunica com outras nos segmentos aos quais se conecta⁶. As bridges executam uma computação conhecida como algoritmo *distributed spanning tree (DST)* para decidir quais não encaminham quadros. O DST permite que uma bridge determine se o encaminhamento introduzirá um ciclo. Em essência, uma bridge não encaminha quadros se descobre que cada segmento que ela liga já contém outra que concordou em encaminhá-los. Após o algoritmo de DST se completar, as bridges que concordam em encaminhar quadros formam um grafo que não contém qualquer ciclo (ou seja, uma *árvore*).

11.13 Comutação (switching)

O conceito de ligação por bridges ajuda a explicar um mecanismo popular, a *comutação (switching)*. Em geral, uma tecnologia de rede é chamada de *comutada (switched)* se o hardware inclui um dis-

⁶ Na maioria das tecnologias, um endereço de hardware especial é reservado para bridges. Por exemplo, as bridges de Ethernet se comunicam usando um endereço multicast reservado exclusivamente para elas.

positivo eletrônico que conecta um ou mais computadores e permite que eles enviem e recebam dados. Mais especificamente, uma *LAN comutada* consiste em um único dispositivo eletrônico que transfere quadros entre muitos computadores.

Fisicamente, um switch se assemelha a um hub – como um hub, um switch consiste em uma única caixa com múltiplas *portas*, cada uma ligada a um único computador. A diferença entre um hub e um switch surge do modo como os dispositivos operam: um hub simula um meio compartilhado único, enquanto um switch simula uma LAN unida através de bridges com um computador por segmento. A Figura 11.10 mostra as conexões conceituais dentro de um switch⁷.

Não é de se surpreender que a vantagem principal de usar uma LAN comutada em vez de um hub é a mesma de usar uma LAN unida por bridges em vez de um segmento único: o paralelismo. Como um hub simula um segmento único compartilhado por todos os computadores, no máximo dois computadores podem se comunicar através de um hub em um determinado tempo. Deste modo, a máxima largura de banda possível de um sistema de hub é R , a taxa em que um único computador pode enviar dados através de um segmento de LAN. Em uma LAN comutada, entretanto, cada computador tem um segmento de LAN simulado para si próprio – o segmento está ocupado apenas quando um quadro estiver sendo transferido de/para o computador. Como resultado, até metade dos computadores conectados a um switch pode estar enviando dados ao mesmo tempo (desde que cada um deles esteja enviando para um dos computadores que não está ocupado com o envio). Deste modo, a largura de banda máxima possível de um switch é $RN/2$, onde R é a taxa em que um determinado computador pode transmitir dados e N é o número total de computadores conectados ao switch.

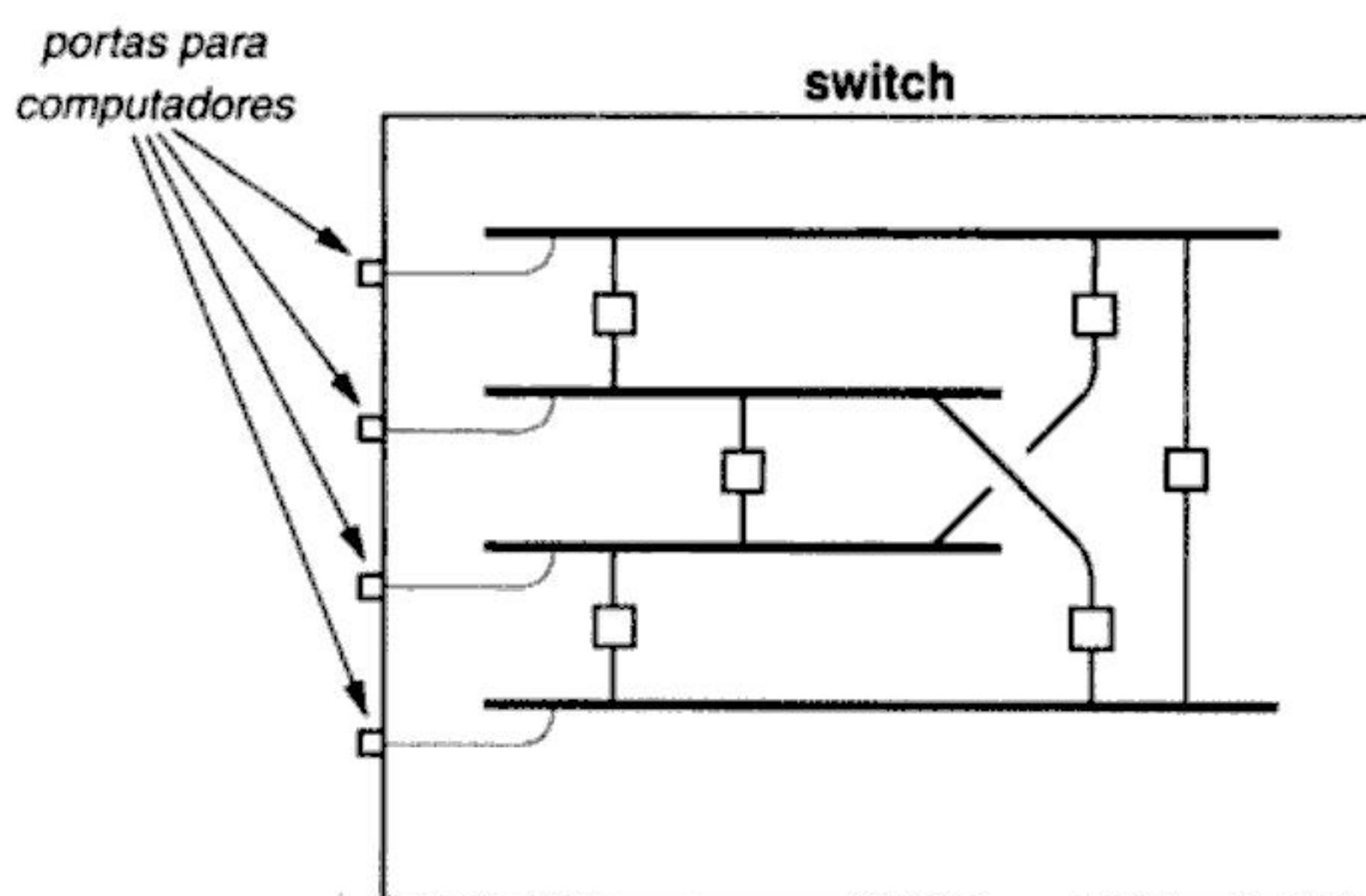


Figura 11.10 O conceito subjacente a uma LAN comutada. O circuitos eletrônicos no switch fornecem a cada computador a ilusão de um segmento de LAN separado conectado a outros segmentos através de bridges.

⁷ Na prática, um switch não é construído por bridges independentes. Em vez disso, um switch contém processadores e uma interconexão central (por exemplo, um cross-bar eletrônico). Um processador examina o endereço de um quadro recebido e então usa a interconexão central para transferi-lo quadro para a porta de saída correta.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

Sumário do Capítulo

- 12.1** Introdução 179
- 12.2** Telefonia Digital 179
- 12.3** Comunicação Síncrona 181
- 12.4** Circuitos Digitais, NIUs e DSU/CSUs 181
- 12.5** Padrões Telefônicos 182
- 12.6** Terminologia DS e Taxas de Dados 183
- 12.7** Circuitos de Capacidade Mais Baixa 184
- 12.8** Circuitos Digitais de Capacidade Intermediária 184
- 12.9** Circuitos de Capacidade Mais Alta 185
- 12.10** Padrões para Portadoras Ópticas 185
- 12.11** O Sufixo C 185
- 12.12** Synchronous Optical NETwork (SONET) 186
- 12.13** O Loop de Assinante Local 187
- 12.14** ISDN 187
- 12.15** Tecnologia da Linha Assimétrica Digital de Assinante 188
- 12.16** Outras Tecnologias de DSL 190
- 12.17** Tecnologia de Modem de Cabo 191
- 12.18** Comunicação Upstream 192
- 12.19** Coaxial de Fibra Híbrido 193
- 12.20** Fibra para o Meio-Fio (Curb) 194
- 12.21** Modems de Head-End e Tail-End 194
- 12.22** Alternativas Sem Fio para Casos Especiais 194
- 12.23** Sistemas de Transmissão por Satélite 195
- 12.24** Resumo 196

Tecnologias para Conexões Digitais de Longa Distância

12.1 Introdução

Os capítulos anteriores descrevem tecnologias de LANs que fornecem comunicação de rede entre um pequeno conjunto de computadores. A limitação fundamental de tecnologias de LANs é a escala: uma única LAN não pode abrigar um número arbitrário de computadores nem a LAN pode conectar computadores em locais arbitrariamente distantes.

Este capítulo começa uma discussão sobre as redes que superam as limitações de tamanho e distância das LANs. O capítulo se concentra em duas peças fundamentais: circuitos de comunicação digital ponto a ponto que formam as conexões básicas de longa distância em uma grande rede e tecnologias que fornecem acesso digital de alta velocidade domicílios e empresas. O próximo capítulo continua a discussão mostrando como essas peças podem ser usadas para formar um grande sistema de rede de comutação de pacotes.

12.2 Telefonia digital

As companhias telefônicas usavam comunicação digital muito antes de as redes de computadores se tornarem importantes¹. A motivação para estudar comunicação digital advém do desejo de tratar de uma grande quantidade de conexões de voz de alta qualidade e longa distância. Os sinais analógicos, base do mecanismo usado nos primeiros sistemas telefônicos, apresentavam problemas em um ambiente de longa distância. Como os sinais elétricos se degradam à medida que viajam por fios de cobre, são necessários amplificadores para melhorar o sinal. Infelizmente, cada amplificador ao longo do caminho distorce ligeiramente o sinal e introduz ruído. A comunicação digital evita o problema do ruído codificando o sinal de áudio original em formato digital, enviando a versão digital através da rede e então recriando o áudio na outra extremidade. A versão digital de um sinal de áudio analógico é chamada de *áudio digital*, e os processos de conversão de um sinal analógico em formato digital é conhecido como *digitalização*.

¹ Os circuitos de voz digital foram usados pela primeira vez em Chicago, Illinois, em 1962.

O hardware para executar a digitalização é um *conversor analógico-digital* (*conversor A-D*)². Um conversor A-D toma um sinal analógico como entrada, faz amostragens do sinal regularmente e computa um número que dá o nível corrente do mesmo (ou seja, a voltagem) no momento da amostra. Deste modo, a digitalização converte um sinal analógico em uma série contínua de números. A Figura 12.1 ilustra o conceito.

Na figura, as linhas verticais representam as amostras e as linhas horizontais pontilhadas representam os valores inteiros possíveis. Em cada amostra, a conversão A-D escolhe o inteiro mais próximo do sinal. Deste modo, a forma digital para o exemplo consiste na seqüência: 0, 2, 4, 4, 7, 1, 1, 1, 1, 1, 2, 4, 4, 4, 4.

Os pesquisadores que investigaram o áudio digital descobriram que a reprodução da voz humana exige um sistema que reproduza freqüências de até 4000 Hz. O *teorema de amostragem* de Nyquist estabelece que se um sinal contínuo é amostrado em uma taxa maior do que duas vezes a freqüência significativa mais alta, o sinal original pode ser reconstruído a partir das amostras. Assim, a voz digitalizada requer que sejam obtidas amostras 8000 vezes por segundo. Isto é, um conversor analógico-digital dentro do sistema telefônico deve amostrar o sinal de um microfone uma vez a cada 125 microsssegundos³. Veremos que a constante de tempo de amostragem é importante na telefonia digital como um todo.

Além de escolher uma taxa de amostragem, quem estiver projetando uma codificação digital deve escolher uma faixa de valores inteiros a ser usada. O compromisso é entre a exatidão e o tamanho dos dados: uma grande faixa de valores permite que o sinal seja reproduzido com maior precisão, mas exige a transmissão de mais bits. Os pesquisadores selecionaram a faixa de 0 a 255 para digitalizar a voz. O esquema de amostragem, parte dos padrões mundiais para telefonia digital, é conhecido como *Pulse Code Modulation (PCM)*. Os valores inteiros produzidos pelo PCM são enviados através de circuitos de longa distância até o destino, onde são convertidos de volta para áudio⁴. Podemos resumir:

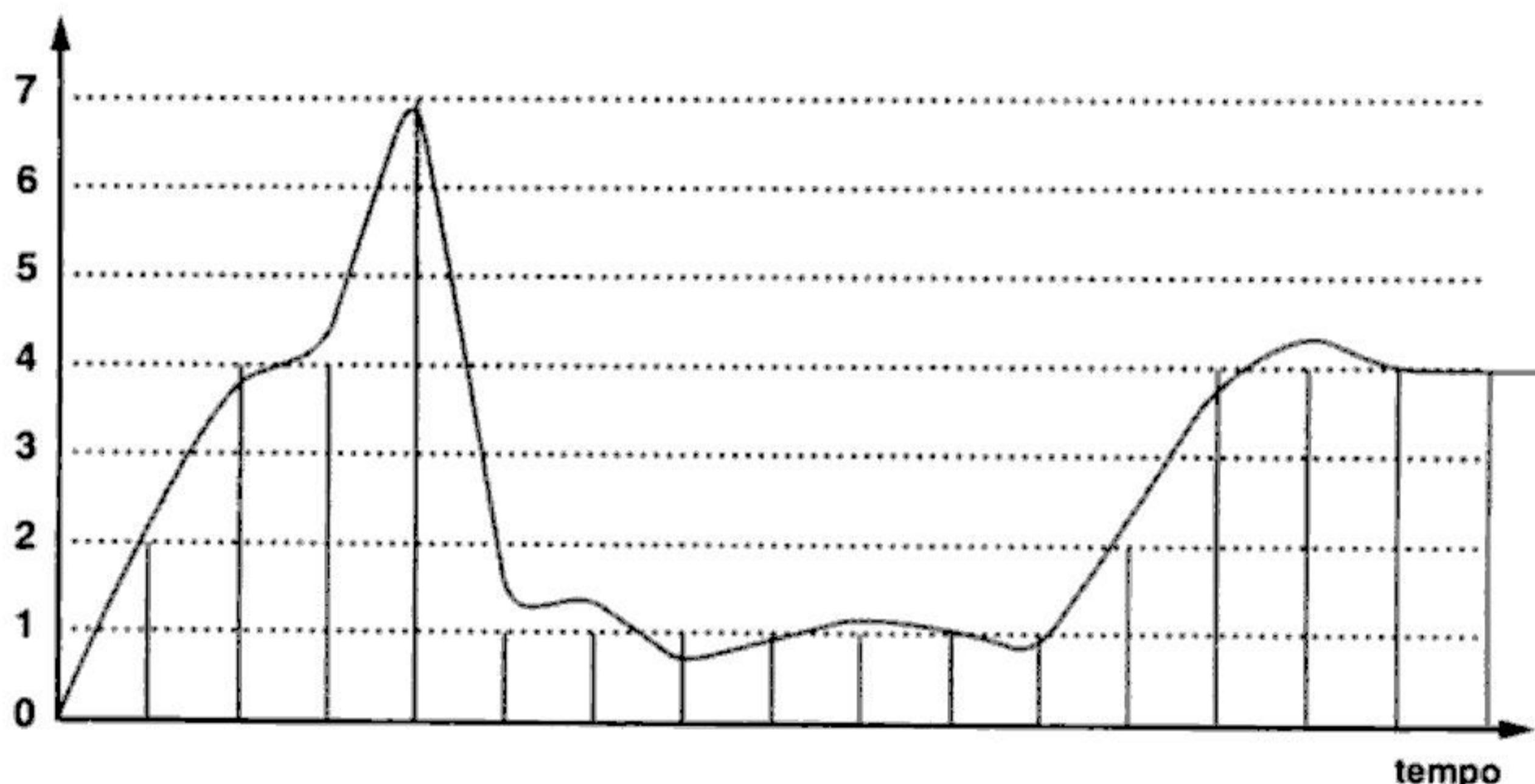


Figura 12.1 Uma ilustração de digitalização usando oito valores. Cada linha vertical representa um valor inteiro escolhido para uma amostra.

² O hardware para recriar um sinal analógico de uma representação digital é chamado de *conversor digital-analógico* (*conversor D-A*).

³ Um microsssegundo é igual a 10^{-6} segundos.

⁴ Uma variante conhecida como *Adaptive Pulse Code Modulation (APCM)* alcança uma faixa mais dinâmica enviando uma seqüência de diferenças em lugar de valores absolutos.

O Pulse Code Modulation é o padrão para codificação digital de áudio usado no sistema telefônico. O PCM obtém amostras de um sinal uma vez a cada 125 microsegundos e converte cada amostra em um inteiro entre 0 e 255.

12.3 Comunicação síncrona

O Capítulo 6 descreveu linhas analógicas que podem ser alugadas das companhias telefônicas e modems que possibilitam o envio de dados digitais através de tais linhas. Porém, as linhas analógicas não são as instalações mais importantes que o sistema telefônico oferece para ligação em redes de dados. Além de modems simples, a indústria telefônica inventou sistemas de comunicação digitais complexos, projetados para transportar informações digitalizadas por longas distâncias. As instalações usadas para voz digitalizada diferem dos sistemas usados para dados porque os sistemas de voz usam tecnologia *síncrona* ou *cronometrada (clocked)*, enquanto a maioria das redes de dados usa tecnologia *assíncrona*. Uma rede síncrona (às vezes chamada de *rede sincronizada ou rede isochronous*) consiste em um sistema projetado para mover dados a uma taxa precisa. Em particular, a rede não diminui a velocidade com os aumentos de tráfego, e os dados emergem da rede na mesma taxa em que entram. Para ver por que a transmissão cronometrada é importante, considere o que poderia acontecer com um sinal de voz digitalizada quando ele fosse transferido através de uma rede não-sincronizada. Com a entrada de mais tráfego na rede, a transmissão de um dado sinal poderia sofrer um aumento no atraso. Deste modo, uma stream de dados que passa pela rede poderia temporariamente desacelerar quando outro tráfego entrasse, e então acelerar novamente quando o tráfego baixasse. Se o áudio de um telefonema digitalizado está atrasado, entretanto, a pessoa que está escutando o telefonema ouvirá o atraso como uma desagradável interferência ou ruído. Mais importante, não existe nenhum modo fácil de recuperação se o fluxo acelera depois de uma lentidão temporária. Uma vez que um receptor começa a tocar amostras digitalizadas que chegam tarde, o receptor não pode “acelerar” a reprodução para alcançar o resto do fluxo. Para evitar tais problemas, o sistema telefônico é cuidadosamente projetado para transmitir informações adicionais junto com os dados digitalizados e para assegurar transmissão contínua. O equipamento receptor usa as informações adicionais para sincronizar seu relógio e assegurar que os dados deixam a rede exatamente na mesma taxa em que entram.

12.4 Circuitos digitais, NIUs e DSU/CSUs

Apesar de serem projetadas para transportar tráfego de voz, as instalações digitais no sistema telefônico têm sido usadas para tráfego de dados. Na verdade, desde os primeiros dias das redes de computadores, as instalações telefônicas digitais têm formado as conexões básicas de longa distância em grandes redes de computadores. As companhias telefônicas alugam os circuitos por uma taxa mensal; é possível alugar um circuito digital ponto a ponto (ou seja, que se estende entre dois edifícios, através de uma grande cidade, ou de um edifício em uma cidade para um edifício em outra). A taxa depende da capacidade do circuito e da distância coberta. Podemos resumir:

Os circuitos digitais alugados de portadoras (carriers) comuns formam as peças fundamentais para as redes de computadores de longa distância. Cada circuito se estende entre dois pontos determinados; a taxa depende da capacidade do circuito e da distância.

Claro, para usar um circuito digital alugado, deve-se concordar em seguir as regras do sistema telefônico, incluindo a adesão aos padrões projetados para transmissão de voz digitalizada. Pode parecer que seguir padrões para informação digitalizada seria trivial, uma vez que os computadores são digitais também. Entretanto, como a indústria de computadores e a indústria de telefonia se desenvolveram independentemente, os padrões para os circuitos digitais do sistema



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

12.5 Padrões telefônicos

Na conexão com um circuito alugado, um DSU deve acomodar os padrões de transmissão digital usados pela companhia telefônica. Nos EUA, os padrões para circuitos de telefone digitais recebem nomes que consistem na letra *T* seguida por um número. Os engenheiros se referem a eles, coletivamente, como a *série T de padrões*. Um dos mais populares é conhecido como T1; muitas empresas usam um circuito T1 para dados⁶.

Infelizmente, os padrões T não são universais. O Japão adotou uma versão modificada da série T de padrões, e a Europa escolheu um esquema ligeiramente diferente. (Os padrões europeus podem ser distinguidos porque usam a letra E.) A Figura 12.3 lista as taxas de dados de três padrões de circuito digitais.

Nome	Taxa de bits	Circuitos de voz	Localização
.	0,064 Mbps	1	
T1	1,544 Mbps	24	América do Norte
T2	6,312 Mbps	96	América do Norte
T3	44,736 Mbps	672	América do Norte
E1	2,048 Mbps	30	Europa
E2	8,488 Mbps	120	Europa
E3	34,368 Mbps	480	Europa

Figura 12.3 Taxas de dados de padrões de circuito digitais populares usados na América do Norte e na Europa.

12.6 Terminologia DS e taxas de dados

Embora as taxas de bit na Figura 12.3 possam parecer números aleatórios, elas são facilmente explicadas. Recorde que as empresas telefônicas projetaram os circuitos digitais para transportar voz. Um único canal de voz exige 64 Kbps (8000 amostras de 8 bits por segundo). A taxa de dados do padrão T1 foi escolhida para permitir que o circuito transporte 24 telefonemas de voz independentes (mais uma pequena quantia de sobrecarga). Um dispositivo é usado para multiplexar as ligações em uma ponta do circuito T1, e outro dispositivo é usado para demultiplexar as ligações na outra extremidade. Por exemplo, imagine dois escritórios telefônicos em uma cidade com um único circuito T1 entre eles. Quando uma ligação deve ser dirigida para o outro escritório, um switch de telefone seleciona um dos canais atualmente não usados no circuito T1 e envia a ligação através daquele canal.

Note que a capacidade dos circuitos não aumenta linearmente com seu número. Por exemplo, o padrão T3 define um circuito com muito mais de três vezes a capacidade do T1. Embora as taxas tenham sido escolhidas para permitir a agregação de ligações, as capacidades maiores não foram selecionadas como múltiplos arbitrários de ligações individuais. Em vez disso, as taxas superiores representam integrais múltiplas de taxas mais baixas. Por exemplo, a taxa T3 é igual a 28 circuitos T1. Deste modo, é possível multiplexar 28 circuitos T1 através de um único circuito T3.

Para entender a motivação para multiplexação, imagine uma empresa de telefones que precisa de múltiplos circuitos T1 entre duas cidades. Ela pode escolher instalar um único circuito T3 e multiplexar até 28 circuitos T1 através dele.

⁶ Além de usar cobre ou fibra óptica para segmentos terrestres, um circuito T1 que atravessa um oceano pode passar através de um satélite.

Para se ser tecnicamente preciso, é necessário distinguir entre os padrões T, que definem o sistema subjacente de transporte, e os padrões que especificam como multiplexar múltiplos telefonemas sobre uma única conexão. Os últimos são conhecidos como *padrões de Digital Signal level* ou *padrões DS*. Os nomes são escritos com as letras *DS* seguidas por um número. Por exemplo, DS1 denota um serviço que pode multiplexar 24 ligações sobre um único circuito. Além disso, como o DS1 define a taxa de dados efetivos, é tecnicamente mais preciso dizer “um circuito executando em velocidade DS1” do que se referir a “velocidade T1”.

Apesar das distinções técnicas menores, os termos T1 e DS1 são freqüentemente trocados. Poucos engenheiros se preocupam em distinguir entre eles. Assim, é provável ouvir alguém citar um circuito como T1 ou se referir a velocidade T1. Podemos resumir:

Os circuitos digitais são classificados de acordo com um conjunto de padrões telefônicos. Dois dos tipos de circuitos mais populares na América do Norte são o T1 e o T3.

12.7 Circuitos de capacidade mais baixa

Atualmente, os circuitos T1 estão entre os mais populares. Porém, um circuito T1 pode ser muito caro para indivíduos ou para empresas pequenas (milhares de dólares por mês). Além disso, muitas empresas não precisam da capacidade de um circuito T1. Em tais casos, é possível alugar um circuito de capacidade mais baixa, conhecido como *T1 fracionário*. Encontram-se disponíveis circuitos fracionários T1 com capacidade muito menor do que 1,544 Mbps (por exemplo, 64 Kbps, 128 Kbps, 9,6 Kbps e 4,8 Kbps). Uma das taxas fracionárias T1 mais populares consiste em um circuito que opera na taxa *DS0* (ou seja, a taxa de uma única chamada de voz, ou 64 Kbps).

A companhia telefônica usa o termo *Time Division Multiplexing (TDM)*⁷ para se referir ao conceito de multiplexão de chamadas de voz digital em um circuito de alta capacidade, ou subdividir um circuito T1 em circuitos de menor capacidade. Para resumir:

Uma empresa que não precisa da capacidade T1 pode economizar alugando um circuito fracionário digital T1. A companhia telefônica usa o termo Time Division Multiplexing para referir-se à tecnologia usada para subdividir um circuito T1. Uma das capacidades fracionárias de T1 mais populares é 64 Kbps.

12.8 Circuitos digitais de capacidade intermediária

E se uma empresa precisa de um circuito com capacidade ligeiramente maior do que o T1, mas somente circuitos T3 estão disponíveis? Como um circuito T3 tem 28 vezes a capacidade de um circuito T1, ele custa substancialmente mais do que o T1. Deste modo, não faz sentido alugar um circuito T3 e usar somente uma pequena fração da sua capacidade. A tecnologia chamada de *multiplexação inversa* foi desenvolvida para tratar de circuitos de capacidade intermediária. A tecnologia permite alugar múltiplos circuitos T1 entre dois pontos e utilizá-los como um único circuito de capacidade mais alta. Um dispositivo eletrônico conhecido como *multiplexador inverso (mux inverso)* é necessário em cada extremidade das linhas. Em um lado, o mux inverso se conecta a um computador. No outro, conecta-se a dois ou mais circuitos digitais. O mux inverso aceita uma stream de dados do computador e envia uma parte dos dados através de cada um dos circuitos digitais. O mux aceita também streams de dados vindos através dos circuitos digitais e os recombina em um stream único. A Figura 12.4 ilustra o conceito.

⁷ Quando usado em referência a circuitos de uma companhia telefônica, o *Time Division Multiplexing* é escrito em caixa alta para distingui-lo do conceito genérico.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

Curiosamente, o tamanho de um quadro SONET depende da taxa de transferência do circuito subjacente. Como mostra a Figura 12.6, um quadro SONET em um circuito STS-1 contém 810 octetos. Entretanto, quando o SONET for usado em um circuito STS-3, cada quadro contém 2430 octetos. De onde esses números surgem? Para entender a diferença, lembre-se de que 125 microssegundos é uma constante fundamental para voz digitalizada porque o PCM exige que se tome uma amostra de 8 bits a cada 125 microssegundos. O SONET usa o tempo para definir o tamanho do quadro. No STS-1, a taxa de transmissão é de 51,840 Mbps, exatamente 6.480 bits são transferidos em 125 microssegundos, o que significa que um quadro consiste em 810 octetos de 8 bits. De forma semelhante, na taxa STS-3, 2.430 octetos podem ser transmitidos em 125 microssegundos. A vantagem principal de fazer o tamanho do quadro depender da taxa de transferência do circuito é que isso possibilita multiplexação síncrona – é justo reter a sincronização ao combinar três streams SONET STS-1 em um stream SONET STS-3.

Embora a maioria das redes de dados use SONET como um esquema de codificação em um circuito ponto a ponto, o padrão fornece mais possibilidades. Em particular, é possível construir uma rede em anel de alta capacidade em rotação contrária usando a tecnologia SONET, que trata de falhas em um único ponto. Cada estação no anel usa um dispositivo conhecido como *mux add/drop*. Além de passar dados recebidos em torno do anel, o mux add/drop pode ser configurado para aceitar dados adicionais de um circuito local e acrescentá-los a quadros que passam através do anel, ou para extraír dados e entregá-los ao computador local. Se o anel for quebrado, o hardware detectará a perda de informações de enquadramento e usará o anel em sentido contrário para reconectar. Para resumir:

Embora o padrão de SONET defina uma tecnologia que pode ser usada para construir uma rede em anel de alta capacidade com múltiplos circuitos de dados multiplexados através das fibras que constituem o anel, a maioria das redes de dados usa SONET apenas para definir o enquadramento e a codificação em um circuito alugado.

12.13 O loop de assinante local

Embora os circuitos alugados de dados forneçam a habilidade de enviar dados através de longas distâncias, outro problema deve ser resolvido antes que as redes de computadores possam se tornar onipresentes: é necessário estender as conexões de velocidade mais alta a residências e empresas. As companhias telefônicas usam o termo *loop local* ou *linha de assinante local* para se referir à conexão entre o Escritório Central (EC) da companhia telefônica e a residência ou empresa; o termo tem sido adotado para se referir às conexões de um provedor de rede até os assinantes individuais. Hoje, a maioria dos loops locais usa sinais analógicos porque foram projetados para o serviço telefônico analógico convencional⁸. A maioria dos assinantes atualmente obtém acesso a redes usando um telefone para discar até um provedor de serviço local.

Embora os modems de dial-up tenham melhorado ao longo dos anos, a largura de banda de voz e a relação entre sinal e ruído das linhas telefônicas limitam a taxa em que bits podem ser enviados. Felizmente, foi inventada uma variedade de novas tecnologias que podem fornecer conexões digitais de alto desempenho para os assinantes.

12.14 ISDN

Um dos primeiros esforços para fornecer serviços digitais em larga escala para os assinantes feito pelas companhias telefônicas sob o nome de *Integrated Services Digital Network (ISDN)*. O ISDN fornece voz e dados digitalizados para assinantes através do cabeamento de loop local convencio-

⁸ O serviço telefônico analógico é freqüentemente chamado de *POTS*, que significa *Plain Old Telephone Service*.

nal. Isto é, o ISDN usa o mesmo tipo de cabeamento de cobre de par trançado que o sistema telefônico analógico.

Do ponto de vista de um assinante, o ISDN oferece três canais digitais separados, designados *B*, *B* e *D* (normalmente escritos *2B+D*). Os dois canais *B*, que operam cada um em uma velocidade de 64 Kbps, são para o transporte de voz digitalizada, dados ou vídeo comprimido; o canal *D*, que opera a 16 Kbps, serve como um canal de controle⁹. Em geral, um assinante usa o canal *D* para solicitar serviços providos pelos canais *B* (por exemplo, fazer um telefonema que usa voz digital). O assinante usa também o canal *D* para administrar uma sessão em andamento ou para terminar uma sessão. Finalmente, os dois canais *B* podem ser combinados ou *unidos* para produzir um único canal com uma taxa de dados efetiva de 128 Kbps.

Os canais *2B+D* são conhecidos como a ISDN *Basic Rate Interface* (BRI). De fato, o ISDN usa uma forma de multiplexação por divisão de tempo para fornecer a ilusão de múltiplos canais de dados viajando através de um único par de fios.

Quando as companhias telefônicas definiram primeiramente o ISDN muitos anos atrás, 64 Kbps parecia rápido se comparado a modems dial-up, que operavam a menos que 10 Kbps. As companhias telefônicas esperavam que os clientes usassem o ISDN para as comunicações digitais locais e de longa distância de forma análoga ao modo que usam o sistema telefônico de voz. Com o passar dos anos, porém, os modems dial-up melhoraram, e foram inventadas tecnologias alternativas que forneceram taxas de dados altas através do loop local com custo mais baixo. Consequentemente, o ISDN agora é uma alternativa cara e com pequena largura de banda.

12.15 Tecnologia da linha assimétrica digital de assinante

Um dos tipos mais impressionantes da nova tecnologia para fornecimento de serviços digitais através de loop local tem o nome de *Digital Subscriber Line* (DSL). Existem diversas variantes, e como os nomes diferem pela primeira palavra, o conjunto é coletivamente chamado pela sigla *xDSL*.

A tecnologia de *xDSL* que talvez seja a mais interessante é conhecida como *Asymmetric Digital Subscriber Line* (ADSL). Do ponto de vista de um assinante, a ADSL fornece a habilidade de enviar e receber informações digitais em alta velocidade. Como o nome indica, entretanto, o serviço é assimétrico. No caso de ADSL, a assimetria deriva da taxa de transferência – a largura de banda disponível é dividida para tornar a taxa de transferência em uma direção muito mais alta do que em outra.

Para entender a motivação para a assimetria, pense sobre como uma pessoa média usa a Internet. A maioria do tráfego é gerada quando a pessoa navega pela Web ou faz download de arquivos. Em ambos os casos, o tráfego que o indivíduo envia para a Internet consiste em pequenas solicitações (por exemplo, alguns bytes de dados). Porém, o tráfego que flui de volta da Internet para o usuário pode conter milhões de bytes de dados (por exemplo, imagens digitalizadas). Para distinguir as duas direções, os profissionais usam o termo *downstream* para se referir a dados fluindo para o usuário, e *upstream* para se referir a dados fluindo a partir do usuário. O loop do assinante pode ser otimizado para tráfego assimétrico se estiver alocando a largura de banda de maneira a fornecer uma taxa de transferência downstream mais alta. Podemos resumir:

A ADSL é uma tecnologia de loop local otimizada para usuários típicos, que recebem muito mais informações do que enviam. Para acomodar tal uso, a ADSL fornece uma taxa de transferência downstream mais alta (ou seja, para o assinante) do que a upstream (ou seja, do assinante para o provedor).

⁹ Embora o usuário tenha um total de 144 Kbps disponível, o sistema subjacente opera a 160 Kbps; os 16 Kbps restantes são consumidos por sincronização e enquadramento.

Do ponto de vista do usuário, a otimização oferece a vantagem de permitir que as páginas da Web sejam mostradas mais depressa do que uma solução simétrica permitiria. Claro, a assimetria torna a ADSL imprópria para conexões que enviam mais dados do que recebem. Por exemplo, uma empresa que tem um catálogo on-line disponível para clientes não se beneficiaria da ADSL, já que tenderia a enviar mais dados do que receber.

Quão rápido a ADSL pode operar? A taxa máxima de downstream atinge um valor espantoso de *6,144 Mbps*, e a taxa máxima upstream atinge *640 Kbps*. Como existe um canal de controle de rede obrigatório que exige *64 Kbps*, a taxa efetiva upstream para dados do usuário é *576 Kbps*.

Embora as taxas de dados sejam surpreendentemente altas, o aspecto mais surpreendente da ADSL advém do cabeamento físico por meio do qual ela alcança tais taxas de dados e o modo que ela usa esse cabeamento. A ADSL não exige quaisquer mudanças no cabeamento de loop local porque é projetada para correr sobre o mesmo cabeamento de par trançado originalmente instalado para o serviço telefônico analógico. Além disso, a ADSL não interrompe o loop local – ela pode executar simultaneamente através dos mesmos fios que o serviço telefônico padrão! Deste modo, a ADSL tem uma vantagem econômica óbvia: as companhias telefônicas podem usá-la para prover serviço digital de alta velocidade sem recabar o loop local. A Figura 12.7 mostra como os modems da ADSL se acoplam ao cabeamento telefônico padrão em paralelo com a existência de equipamento telefônico analógico. Como o serviço é assimétrico, os modems usados nas duas pontas de uma linha diferem ligeiramente.

Como a ADSL alcança taxas altas de dados sobre par trançado? Os pesquisadores primeiramente observaram que um esquema como a ADSL poderia ser possível porque muitos loops locais acomodam sinais em freqüências mais altas do que aquelas usadas pelo sistema telefônico. A solução ADSL é complexa porque não há dois loops locais com características elétricas idênticas. Em vez disso, a habilidade de transportar sinais depende da distância, da medida de cabeamento usado e do nível de interferência elétrica. Deste modo, os projetistas não podem escolher um conjunto particular de freqüências portadoras ou técnicas de modulação que funcionem em todos os casos. Por exemplo, considere dois assinantes que vivem em partes diferentes de uma cidade. Se a linha telefônica que está indo ao primeiro assinante passa próxima a uma estação de rádio comercial, o sinal da estação cau-

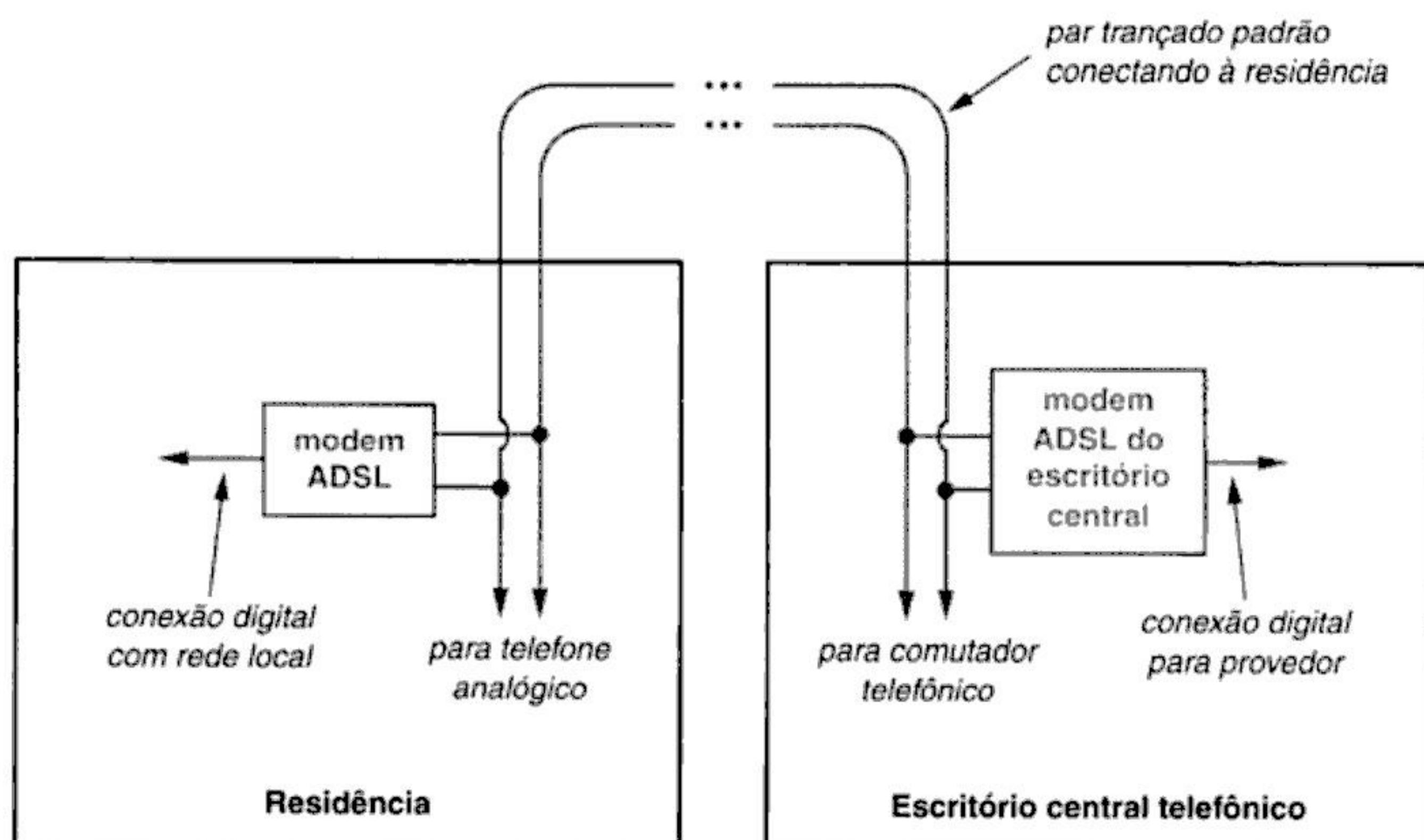


Figura 12.7 Modems de ADSL conectados ao cabeamento local existente. Os modems podem usar um par de fios simultaneamente com serviço telefônico analógico.

sará interferência na freqüência que a estação usa. Se o segundo assinante não vive próximo à mesma estação de rádio, a freqüência que a estação de rádio usa pode funcionar bem para dados naquele linha do assinante. Porém, a segunda linha pode sofrer interferência em outra freqüência.

Para acomodar as diferenças de características de loop local, a ADSL é adaptável. Ou seja, quando os modems ADSL são ligados, examinam a linha entre eles para descobrir suas características e, então, concordam em se comunicar usando técnicas ótimas para a linha. Em particular, a ADSL usa um esquema conhecido como *modulação Discrete Multi Tone (DMT)*, que combina multiplexação por divisão de freqüência e técnicas de multiplexação inversa.

A multiplexação por divisão de freqüência no DMT é implementada dividindo a largura de banda em 286 freqüências separadas ou *subcanais*¹⁰, com 255 freqüências usadas para transmissão downstream de dados, 31 usadas para transmissão upstream e 2 usadas para informações de controle. Conceitualmente, existe um “modem” separado executando em cada subcanal, com sua própria portadora modulada. As portadoras são espaçadas em intervalos de 4,1325 KHz para evitar que os sinais interfiram um no outro. Além disso, para garantir que suas transmissões não interfiram com sinais de telefonia analógica, a ADSL evita usar a largura de banda abaixo de 4 KHz. Quando a ADSL inicia, as pontas examinam as freqüências disponíveis para determinar que sinais passam e que sinais experimentam interferência. Além de selecionar freqüências, as duas pontas avaliam a qualidade do sinal em cada freqüência e usam a qualidade para selecionar um esquema de modulação. Se uma freqüência em particular tem uma relação sinal-ruído alta, a ADSL seleciona um esquema de modulação que codifica muitos bits por baud; se a qualidade em uma determinada freqüência é baixa, a ADSL seleciona um esquema de modulação que codifica menos bits por baud. Podemos resumir:

Para alcançar taxas de bit altas através de cabeamento de par trançado convencional, a ADSL usa uma tecnologia adaptável, em que um par de modems examina muitas freqüências na linha entre eles e seleciona freqüências e técnicas de modulação que forneçam resultados ótimos naquela linha.

O resultado da adaptação é uma tecnologia robusta que pode se adaptar a várias condições de linha automaticamente. Do ponto de vista de um usuário, adaptação tem uma propriedade interessante: a ADSL não garante uma taxa de dados. Em vez disso, a ADSL pode garantir somente que fará tanto quanto as condições da linha permitirem que suas técnicas operem. Deste modo, a taxa downstream varia de 32 Kbps a 6,4 Mbps e a upstream varia de 32 a 640 Kbps.

12.16 Outras tecnologias de DSL

Além de ADSL, foram desenvolvidas outras tecnologias de DSL. Cada uma usa a largura de banda para alcançar uma meta ligeiramente diferente. Deste modo, cada uma apresenta vantagens para alguns aplicativos. Por exemplo, a *Symmetric Digital Subscriber Line (SDSL)* fornece taxas de bit simétricas em ambas as direções. Como descrito anteriormente, a maioria dos indivíduos que usa redes de computadores segue um padrão assimétrico. Diversas companhias pequenas seguem o mesmo padrão – geralmente usam a rede para obter informações. Porém, as empresas que fornecem informações para outras tendem a ter exatamente a assimetria oposta – exportam mais dados que importam. Infelizmente, poucas companhias telefônicas oferecem ADSL com a direção de assimetria invertida. Deste modo, as empresas que exportam informações podem preferir a SDSL. Além disso, como a SDSL usa um esquema de codificação diferente da ADSL, ela pode operar através de loops locais para os quais a ADSL é imprópria. Assim, algumas companhias telefônicas escolhem oferecer serviço de SDSL, em vez de ADSL.

¹⁰ A terminologia surge porque a capacidade de uma conexão de ADSL pode ser dividida em “canais” de 1,544 Mbps conectados, cada um, a um circuito T1.

Outro serviço de DSL é conhecido como *High-Rate Digital Subscriber Line (HDSL)*. O HDSL fornece uma taxa de bits DS1 (ou seja, 1,544 Mbps) nas duas direções. Uma das desvantagens do HDSL é uma limitação de distância em loops locais. Outra desvantagem advém dos requisitos de cabeamento. Diferentemente da ADSL, que usa um único par trançado, o HDSL exige dois pares trançados independentes. Para superar a desvantagem de cabeamento, uma variante conhecida como *HDSL2* foi proposta, e esta executa sobre dois fios¹¹.

Uma das vantagens do HDSL advém de sua tolerância a modificações no loop local feitas para o sistema telefônico. Em particular, o HDSL pode ser usado em um loop que inclui uma *bridge tap* telefônica (algumas tecnologias de DSL não podem). Além disso, como sua taxa de transferência é compatível com um circuito T1, mover dados entre um circuito T1 e HDSL é simples.

Outra vantagem de HDSL surge da sua habilidade de tolerar falhas. A tecnologia foi projetada de forma que se um dos dois pares trançados falha, os modems não falham completamente, eles continuam a operar à metade da taxa máxima de transferência. A tolerância à falha é especialmente atrativa para as empresas, pois é freqüentemente melhor ter uma conexão lenta do que nenhuma conexão.

Outra variante de DSL estudada pode fornecer throughput muito mais alto. Conhecida como *Very-high bit rate Digital Subscriber Line (VDSL)*, a tecnologia pode alcançar uma taxa de dados de até 52 Mbps. Alternativas que oferecem 13 Mbps e 26 Mbps também foram investigadas.

Embora essas taxas altas possam ser alcançadas por meio de um par de cobre trançado, a VDSL não pode ser usada no cabeamento existente entre a Estação Central e os assinantes telefônicos porque as distâncias são muito longas. Então, a VDSL exige pontos de concentração intermediários (por exemplo, um em cada bairro), com fibra óptica conectando os pontos de concentração de volta para a EC. Em terminologia VDSL, um ponto de concentração é chamado de *Optical Network Unit (ONU)*. Como as versões de VDSL com taxas de dados inferiores correm pelo cobre através de distâncias mais longas, elas não exigem que pontos de concentração fiquem tão perto do assinante. Portanto, taxas de dados mais baixas exigem menos pontos de concentração para cobrir uma dada área geográfica.

12.17 Tecnologia de modem de cabo

As seções anteriores examinaram tecnologias que entregam informações digitais através do cabeamento de par trançado que forma o loop local do sistema telefônico analógico. Esta seção considera o uso de um esquema de cabeamento alternativo que pode entregar taxas de bit muito mais altas.

A motivação primária para considerar alternativas para o loop local telefônico advém das limitações inerentes. O problema principal reside nas características elétricas do cabeamento de par trançado. Embora tecnologias como ADSL possam alcançar taxas de bit muito mais altas que modems dial-up, o cabeamento impõe um limite máximo na rapidez com que os dados podem ser transferidos. Além disso, a falta de proteção torna o cabeamento suscetível a interferências, o que pode prejudicar substancialmente o desempenho para alguns assinantes.

Em um esforço para superar as limitações do cabeamento de par trançado, os pesquisadores investigaram tecnologias com e sem fio para uso no loop local. Uma tecnologia alternativa distingue-se como particularmente atraente porque oferece velocidade mais alta do que cabeamento telefônico, é menos suscetível à interferência eletromagnética e não exige uma infra-estrutura completamente nova: a televisão a cabo¹². Além disso, muitas áreas residenciais já têm uma instalação de TV a cabo disponível.

Um sistema de TV a cabo tem quase todas as facilidades necessárias para o envio de informações digitais downstream em alta velocidade. O meio consiste em cabo coaxial, que tem alta capacida-

¹¹ Como o tráfego upstream e downstream compartilha um único par de fios, o HDSL2 é às vezes chamado de SHDSL.

¹² Formalmente, a tecnologia é conhecida como *Community Antenna TeleVision (CATV)*.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

proximidade geográfica (por exemplo, em um bairro). O grupo compartilha uma freqüência portadora, com somente um assinante no grupo capaz de receber um pacote em um dado momento. A largura de banda mais alta da fibra possibilita multiplexar vários grupos independentes por meio das linhas de tronco.

Como a ADSL, os modems de cabo foram projetados para fornecer taxas mais altas downstream do que upstream. A taxa de dados para entrega upstream pode atingir de 1,5 a 2,0 Mbps. Porém, como os dados de múltiplos assinantes devem ser multiplexados em uma largura de banda de 6 MHz, há um declínio da taxa efetiva quando muitos usuários transmitem dados simultaneamente.

Para resumir:

Para usar Hybrid Fiber Coax, as companhias de cabo devem mudar grande parte da infra-estrutura central. As linhas de tronco devem ser substituídas por fibra óptica e todos os amplificadores devem ser modificados para operar em ambas as direções. Porém, o sistema pode usar os circuitos de alimentação coaxial existentes para alcançar assinantes individuais.

12.20 Fibra para o meio-fio (curb)

Além do HFC, as empresas de cabo têm explorado outras tecnologias. Uma alternativa é conhecida como *Fiber to the Curb (FTTC)*. Como o nome indica, a FTTC é semelhante ao HFC porque usa fibra óptica para troncos de alta capacidade. A idéia é levar fibra óptica até perto do assinante e então usar fio de cobre para os circuitos alimentadores.

A FTTC difere do HFC porque usa dois meios em cada circuito alimentador para permitir que o sistema de cabo forneça um serviço adicional. Isto é, a companhia de cabo deve levar um fio adicional a cada lar para que o HFC seja usado. O primeiro circuito usa cabo coaxial existente para entregar vídeo interativo. O segundo usa par trançado, que pode ser usado para transportar voz.

12.21 Modems de head-end e tail-end

Tecnologias de loop local populares como a ADSL ou modems a cabo são designadas para fornecer maior desempenho do escritório central para o assinante do que do assinante para o escritório central. Para alcançar taxas de dados assimétricas, os modems usados nos dois lugares devem ser diferentes. O termo *head-end modem* se refere ao modem usado no escritório central, e o termo *tail-end modem* se refere ao modem usado pelos assinantes. O head-end modem usado para transmitir dados através do cabo segue o padrão do *Cable Modem Termination System (CMTS)*.

Além dos padrões básicos para comunicação, fornecedores de Internet a cabo precisam de padrões que permitam aos assinantes configurar interativamente seu acesso à rede ou demandar serviços especiais (como filmes pay-per-view). Para tais necessidades, muitos sistemas de cabo implementam o *Data Over Cable System Interface Standard (DOCSIS)*. DOCSIS não especifica o que pode ser demandado e como as contas e cobranças devem ser administradas; ele especifica como um assinante envia pedidos e recebe respostas.

12.22 Alternativas sem fio para casos especiais

Embora tecnologias como ADSL ou HFC possam entregar serviços digitais para a maioria dos assinantes, elas não abrangem todas as circunstâncias. Os problemas primários surgem em áreas rurais. Por exemplo, imagine uma fazenda a muitos quilômetros da cidade mais próxima ou uma vila remota. O cabeamento de par trançado usado para entregar serviço telefônico para tais localidades excede à distância máxima para tecnologias como ADSL. Além disso, as áreas rurais são as que têm menor probabilidade de possuir um serviço de TV a cabo.

Tecnologias como ADSL têm restrições técnicas adicionais no tipo de linha que podem usar. Por exemplo, pode ser impossível usar freqüências altas nas linhas telefônicas que contêm loading coils, taps ou repetidores. Além disso, o cabeamento de cobre deve ter uma medida suficiente para que os sinais não sejam atenuados. Deste modo, mesmo em áreas onde uma tecnologia de loop local funcione para a maioria dos assinantes, a tecnologia pode não funcionar para todas as linhas.

Para tratar de casos especiais, tem sido explorada uma variedade de tecnologias alternativas de loops locais alternativas. Por exemplo, é possível usar tecnologias sem fio. Serviços de dados a baixas velocidades usam tecnologias similares àquelas usadas por telefones celulares digitais. O IEEE desenvolveu uma nova tecnologia sem fio de maior velocidade, que pode fornecer até 155 Mbps: o padrão IEEE 802.16. A tecnologia recebeu o nome de WiMAX por um conjunto de vendedores. A WiMAX é classificada como uma *Rede de Área Metropolitana* (*Metropolitan Area Network, MAN*) porque ela se expande a distâncias maiores do que uma LAN, mas menores do que uma WAN. A WiMAX pode ser usada como um substituto de loop local ou como uma tecnologia que fornece acesso de maior velocidade entre um bairro e o escritório central. Por exemplo, ISPs podem usar WiMAX dentro de uma cidade para ligar pontos de concentração afastados.

12.23 Sistemas de transmissão por satélite

Outra alternativa usa satélites de comunicação digital. Inicialmente, tais satélites eram usados pelas empresas de telecomunicação, primariamente como uma alternativa para linhas terrestres. Então, eram usados em situações nas quais as linhas terrestres eram caras ou impraticáveis, como através de oceanos ou em áreas montanhosas. Mais importante, os primeiros satélites forneciam comunicação de ponto a ponto (por exemplo, de uma estação fixa nos Estados Unidos para uma estação fixa na Europa). Quando se tornou óbvio que companhias comerciais podiam construir, utilizar e operar satélites de comunicação, surgiu a pergunta: os satélites podem ser usados como uma tecnologia de loop local? Caso possam, seriam apropriados para casos específicos ou conseguiram fornecer uma estrutura para propósitos gerais?

Por um lado, os satélites oferecem vantagens: um sistema de satélite tem mais largura de banda do que uma conexão discada e pode alcançar uma localidade geográfica arbitrária. Por outro lado, a órbita correta de um satélite geossincrônico acarreta grandes esperas. Mais importante, o equipamento da estação fixa necessário para transmitir o sinal ao satélite (conhecido como um *uplink*) é ao mesmo tempo caro e grande, tornando-o inapropriado para indivíduos ou pequenos negócios – apenas grandes corporações podem bancar tais dispositivos.

Para desenvolver um sistema de loop local usando satélites, duas inovações eram necessárias:

- Em vez de tratar satélites como sistemas de comunicação de ponto a ponto, um mecanismo de difusão (*broadcast*) foi criado.
- Em vez de uma grande e cara estação fixa, um caminho de transmissão de *uplink* alternativo foi utilizado.

O mecanismo de broadcast via satélite usa o mesmo princípio descrito no capítulo 9: o satélite transmite cada pacote e todas as estações sintonizadas no satélite recebem uma cópia. Para garantir que os pacotes cheguem somente ao destino previsto, cada estação é designada com um *endereço* único, e a estação filtra os pacotes entrantes da mesma forma que uma placa de interface de LAN os filtra. Isto é, embora o canal do satélite seja compartilhado, uma determinada estação aceita somente pacotes enviados ao endereço da estação; os demais pacotes são descartados.

Claro, a transmissão via satélite é mais complicada do que a transmissão através de uma LAN. Em particular, um satélite pode conter múltiplos transmissores, cada um operando num canal separado (pela freqüência). Então, um satélite de *broadcast* pode usar a divisão de freqüências por multiplexão para alcançar maior largura de banda, e usar a divisão de tempo por multiplexão para per-

mitir que múltiplos receptores compartilhem a largura de banda. Para o propósito do livro, entretanto, é suficiente pensar no satélite como se oferecesse um único canal compartilhado.

Para resolver o problema do *uplink*, os fornecedores de satélites se concentraram nos requerimentos da comunicação assimétrica descrita anteriormente. Eles observaram que um satélite pode ser usado como um caminho de downstream de alta capacidade, enquanto um caminho de baixa velocidade pode ser usado para o tráfego de upstream. Em particular, fornecedores que oferecem conexões de Internet via satélites de *broadcast* dependem de conexões discadas convencionais para o tráfego de upstream. O ponto importante é:

A tecnologia de satélites de broadcast estende a noção de entrega assimétrica para usar dois mecanismos subjacentes diferentes. Embora um satélite de broadcast seja usado para o tráfego de downstream, o tráfego de upstream viaja através de uma rede de capacidade mais baixa, como uma conexão de linha telefônica discada convencional.

Então, para usar uma tecnologia de broadcast via satélite, cada assinante deve ter uma conexão de telefone, bem como uma antena (uma parabólica receptora) e um computador. O distribuidor do satélite fornece um dispositivo de hardware que interconecta a antena, o computador e a linha telefônica. O dispositivo aceita pacotes da antena e os entrega ao computador. Ele também aceita pacotes do computador e os envia da linha telefônica. Para fazê-lo, o dispositivo possui um modem de linha discada, e é configurado para saber como discar e enviar pacotes. Para fazer todo o sistema funcionar, o roteamento é arranjado com muito cuidado, garantindo que o tráfego enviado para qualquer assinante de satélite seja repassado ao fornecedor do satélite, que então transmite o pacote.

12.24 Resumo

As companhias telefônicas foram pioneiras na comunicação digital de longa distância para fornecer conexões de voz de longa distância e de alta qualidade entre cidades. É possível alugar circuitos digitais ponto a ponto das companhias telefônicas para uso privado; o custo depende da capacidade e da distância coberta. A maioria das Redes de Longo Alcance usa os circuitos digitais alugados para fornecer comunicação de longa distância.

As portadoras comuns estabeleceram padrões para representação e taxas de dados de circuitos digitais; a maioria dos padrões foi projetada para acomodar múltiplas streams de tráfego de voz digitalizada. Um dispositivo de DSU/CSU deve estar acoplado a cada extremidade de um circuito digital para traduzir entre a representação digital que os computadores usam e a representação usada pela companhia telefônica.

Na América do Norte, muitas redes de computadores usam circuitos digitais T1 ou T3. Um circuito T1 tem uma taxa de dados de 1,544 Mbps, e um circuito T3 tem uma taxa de dados de 44,736 Mbps. Há também circuitos de capacidade mais alta, que usam os padrões *Synchronous Transport Signal (STS)*. Como as taxas altas de dados exigem o uso de fibra óptica como meio, um conjunto paralelo de padrões para *Optical Carrier (OC)* define os sinais ópticos usados. Por exemplo, um circuito OC-3 opera a 155,520 Mbps.

Além de circuitos ponto a ponto que atravessam longas distâncias, são necessários mecanismos que entreguem dados a empresas ou residências individuais. As companhias telefônicas têm investigado várias tecnologias de *loop local* (termo usado para descrever as conexões entre uma estação central e um assinante). Em particular, a *Asymmetric Digital Subscriber Line (ADSL)* se sobressai por fornecer comunicação digital de alta velocidade através do cabeamento de par trançado existente usado para serviço telefônico analógico. A ADSL é especialmente atraente porque permite que o serviço telefônico convencional opere na linha simultaneamente.

As empresas que oferecem serviço de televisão a cabo têm usado tecnologia de cabo para transmitir informações digitais até os assinantes. Um dos obstáculos principais surge da infra-estrutura

de cabo existente, projetada para transportar informações somente em uma direção. Deste modo, as empresas de cabo estão substituindo partes significativas de sua infra-estrutura para que seja possível a comunicação bidirecional.

As empresas de cabo desenvolveram a tecnologia de *Hybrid Fiber Coax (HFC)* para fornecer comunicação bidirecional. Embora o HFC use fibra óptica no lugar das linhas de tronco atuais, o HFC pode usar cabo coaxial já existente para conexões com assinantes individuais. Uma tecnologia alternativa conhecida como *Fiber To The Curb (FTTC)* exige que as empresas de cabo substituam linhas de tronco por fibra e acrescentem cabeamento de par trançado para cada assinante.

Fornecedores de satélite oferecem um mecanismo de loop local que estende a entrega assimétrica para usar hardware assimétrico. Embora o satélite seja usado para o tráfego de downstream, o tráfego de upstream viaja por outro caminho, geralmente uma conexão de linha discada.

Exercícios

- 12.1 Descubra se sua instituição aluga circuitos digitais. Caso positivo, a que localizações se conectam?
- 12.2 Contate uma portadora comum para descobrir as taxas de aluguel mensais para uma linha T1, uma linha T3 e uma linha OC-3 de sua cidade até outra cidade no outro lado do país.
- 12.3 Leia mais sobre ISDN. Por que é chamado de tecnologia *comutada*?
- 12.4 Considere uma página da Web que contenha um total de 6 megabytes de imagens. Quanto tempo levará para enviar os dados através de um circuito T1, um circuito T3, um circuito OC-3, um circuito OC-12 e um circuito OC-48? (Ignore toda a sobrecarga de protocolo.)
- 12.5 Leia sobre a tecnologia de transmissão direta por broadcast via satélite. Qual é a desvantagem principal do serviço digital obtido via satélite?
- 12.6 Além dos sinais enviados por outro modem ADSL, a ADSL deve tolerar os sinais enviados pelo sistema telefônico analógico. Em particular, o modem deve tolerar a corrente elétrica usada para tocar um telefone. Leia sobre o serviço telefônico analógico para descobrir que voltagem é usada.
- 12.7 Leia sobre *Adaptive Pulse Code Modulation (APCM)*. Qual é a vantagem principal de APCM sobre PCM?
- 12.8 Suponha que você receba dois circuitos digitais e seja informado de que um deles usa um satélite e o outro, apenas fibra óptica. Que experimento você poderia executar para determinar qual usa fibra?
- 12.9 Na pergunta anterior, seu método funcionaria mesmo se os dois circuitos tivessem taxas de bit diferentes (por exemplo, um circuito T1 e um circuito OC-3)?



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

A figura mostra cada endereço como um par de inteiros decimais. Um computador conectado à porta 6 no switch de pacotes 2 recebe o endereço atribuído [2, 6]. Na prática, um endereço é representado como um valor binário único, com alguns bits desse valor usados para representar a primeira parte do endereço e outros para representar a segunda parte. Como cada endereço é representado como um valor binário único, os usuários e programas aplicativos podem tratá-los como um inteiro único – não precisam saber que endereços são designados hierarquicamente.

13.7 Encaminhamento next-hop

Um switch de pacotes deve escolher um caminho de partida através do qual encaminha cada pacote. Se o pacote é destinado a um dos computadores acoplados diretamente, o switch encaminha-o para o computador. Se o pacote é destinado a um computador acoplado a outro switch de pacotes, ele deve ser encaminhado através de uma das conexões de alta velocidade que levam ao switch. Para fazer a escolha, um switch de pacotes usa o endereço de destino armazenado no pacote.

Um switch de pacotes não mantém informações completas sobre como alcançar todos os destinos possíveis. Em vez disso, um determinado switch tem informações sobre o próximo lugar (*hop*) para enviar um pacote de maneira que este, mais cedo ou mais tarde, alcance seu destino. Chamado de *encaminhamento ao próximo hop (next-hop forwarding)*, o conceito é análogo à forma que as linhas aéreas listam seus vôos. Suponha uma linha aérea em que um passageiro que quer viajar de São Francisco a Miami descobre que o único itinerário disponível envolve três vôos: o primeiro de São Francisco até Dallas, o segundo de Dallas até Atlanta e o terceiro de Atlanta até Miami. O último destino permanece o mesmo ao longo da viagem: Miami. Porém, o próximo hop muda em cada aeroporto. Quando o passageiro estiver em São Francisco, o próximo hop será Dallas. Quando o passageiro estiver em Dallas, o próximo hop será Atlanta, e quando o passageiro estiver em Atlanta, o próximo hop será Miami. A Figura 13.4 mostra o encaminhamento ao próximo hop em uma rede com comutação de pacotes.

Como mostra a figura, as informações do próximo hop podem ser organizadas em uma tabela. Cada entrada na tabela lista um destino e o próximo hop usado para alcançá-lo. Quando está encaminhando um pacote, o switch extrai o destino do pacote, procura na tabela uma entrada que combine com o destino e então envia o pacote para o próximo hop especificado na entrada. O exemplo de tabela mostra como o switch de pacotes 2 encaminha pacotes. Quando encontra um pacote des-

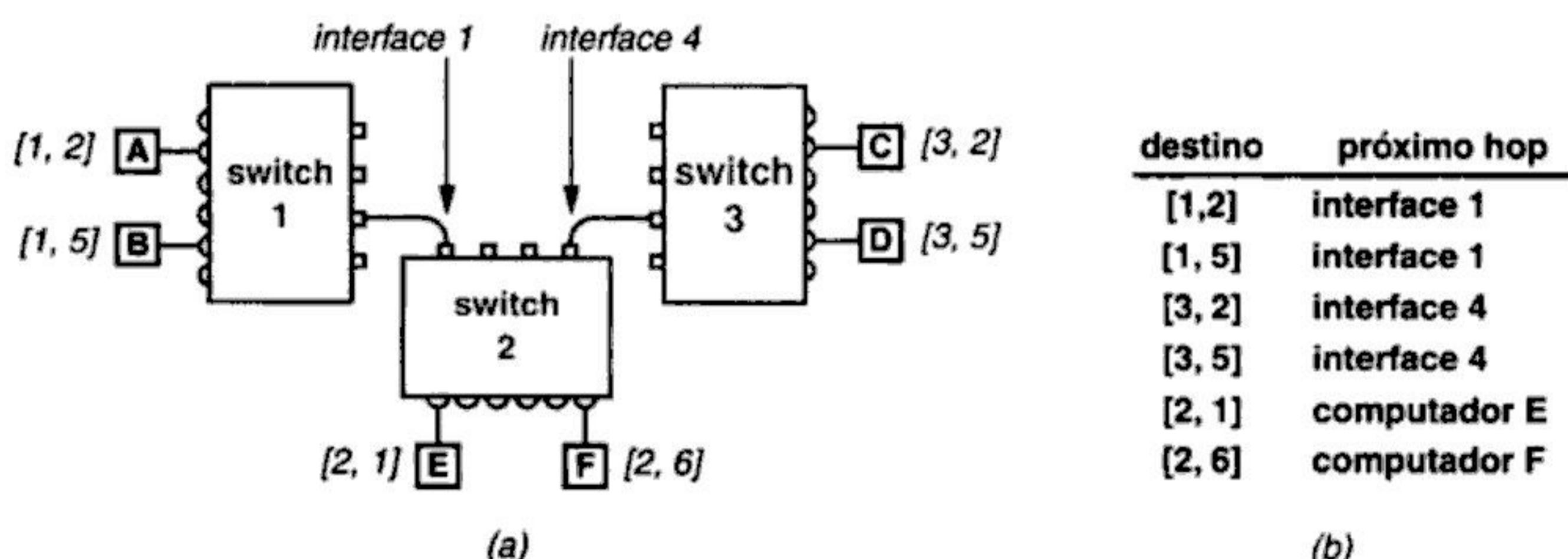


Figura 13.4 (a) Uma rede que consiste em três switches de pacote e (b) as informações de encaminhamento para o próximo hop conforme encontradas no switch 2. Cada switch tem informações do próximo hop diferentes.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

maior do que a soma de todos os pesos juntos para qualquer caminho no grafo. Uma maneira de gerar um valor *infinito* consiste em adicionar um à soma de todos os pesos em todas as arestas. Permitir que sejam atribuídos pesos arbitrários às arestas de um grafo significa que um algoritmo pode ser usado com medidas diferentes de distância. Por exemplo, algumas tecnologias de WAN medem distância contando o número de switches de pacote ao longo de um caminho. Para usar o algoritmo com tais tecnologias, a cada aresta no grafo é atribuído um peso *1*. Em outras tecnologias de WAN, são atribuídos pesos para refletir a capacidade das conexões subjacentes. Finalmente, algumas tecnologias de WAN atribuem pesos para implementar políticas administrativas.

13.14 Cálculo de rota distribuída

O Algoritmo 13.1 mostra como uma tabela de roteamento pode ser calculada depois de serem codificadas informações sobre uma rede em um grafo. A técnica alternativa é um *cálculo de rota distribuída*, em que cada switch de pacotes calcula sua tabela de roteamento localmente e então envia mensagens através da rede para switches de pacote vizinhos, informando-os do resultado.

As redes que usam cálculo de rota distribuída fazem com que cada switch de pacotes envie periodicamente suas informações de roteamento para os vizinhos (por exemplo, a cada poucos segundos). Depois de um período inicial de partida, cada switch de pacotes aprende os caminhos mais curtos para todos os destinos – um algoritmo distribuído produz as mesmas tabelas de roteamento de próximo hop como o Algoritmo 13.1. As mensagens periódicas contínuas permitem que a rede se adapte, caso um switch de pacotes individual ou um link de comunicação individual falhem. Depois de uma falha, um switch de pacotes pára de receber atualizações do hardware que falhou. O switch continua recebendo atualizações de vizinhos que estão funcionando e, se existe um caminho alternativo, ele pode modificar sua tabela de roteamento para evitar o hardware que falhou.

13.15 Roteamento de vetor de distância

Um dos algoritmos mais conhecidos para cálculo de rota distribuída é o *algoritmo de vetor de distâncias (distance-vector algorithm)*⁶. Como no Algoritmo 13.1, a cada link na rede é atribuído um peso, e a *distância* para um destino é definida com a soma dos pesos ao longo do caminho até o destino. Como os valores necessários para o cálculo precisam ser armazenados, um campo adicional é acrescentado a cada entrada da tabela de roteamento. O campo adicional contém a distância até o *destino* (conforme indicado pelo campo da entrada), através do caminho indicado pelo *próximo hop* (conforme campo da mesma entrada).

Um switch de pacotes envia periodicamente informações de roteamento através da rede para seus vizinhos. Cada mensagem contém pares (*destino, distância*). Quando uma mensagem chega a um switch de pacotes vindo do vizinho *N*, o switch de pacotes examina cada item e muda sua tabela de roteamento caso o vizinho tenha para algum destino um caminho menor do que o que o switch de pacotes vinha usando. O Algoritmo 13.1 especifica precisamente como as rotas são atualizadas.

13.16 Roteamento por estado de link (SPF)

Existe uma forma alternativa de cálculo de rota distribuída que usa uma versão do Algoritmo 13.1, em vez de um algoritmo de vetor de distância. Formalmente, o algoritmo alternati-

⁶ O nome surge porque as mensagens enviadas de um switch de pacotes até outros contêm pares de valores que especificam um destino e uma distância até esse destino.

Algoritmo 13.1

Dado:

um grafo com um peso não-negativo atribuído a cada aresta e um nó de origem atribuído.

Calcular:

a distância mais curta do nó de origem até todos os outros nós e uma tabela de roteamento de próximo hop

Método:

Incialize o conjunto S para conter todos os nós, com exceção do nó de origem;

Incialize o vetor D de forma que $D[v]$ seja o peso da aresta da origem para v, se existir tal aresta, e *infinito*, caso contrário;

Incialize as entradas de R de forma que $R[v]$ seja atribuído v se existir uma aresta da origem até v, e zero, caso contrário;

enquanto (conjunto S não está vazio) {

escolha um nó u de S tal que $D[u]$ seja mínimo;

se ($D[u]$ é *infinito*) {

erro: não existe nenhum caminho para nós em S; termine;

}

apague u do conjunto S;

para cada nó v tal que (u, v) seja uma aresta {

se (v ainda está em S) {

$c = D[u] + \text{peso}(u, v);$

se ($c < D[v]$) {

$R[v] = R[u];$

$D[v] = c;$

}

}

}

}

Algoritmo 13.1 Uma variante de algoritmo de Dijkstra que computa R , uma tabela de roteamento de próximo hop e D , a distância até cada nó a partir do nó de origem especificado.

vo é conhecido como *link-state* ou *link-status routing*. Informalmente, o esquema se tornou conhecido como *Shortest Path First*, ou roteamento *SPF*⁷.

Embora switches de pacote que usam SPF enviem mensagens através da rede, as mensagens não contêm informações de tabelas de roteamento. Em vez disso, cada mensagem transporta o status de um link entre dois switches de pacote (por exemplo, o link entre o switch 5 e o switch 9 está ativo), e a mensagem é enviada por broadcast para todos os switches. Cada switch coleciona mensagens de status recebidas e as usa para construir um grafo da rede. O switch então usa o Algoritmo 13.1 para produzir uma tabela de roteamento tendo a si próprio como origem.

⁷ O nome é um pouco enganoso porque todos os algoritmos de roteamento encontram caminhos mais curtos.

Algoritmo 13.2

Dado:

uma tabela de roteamento local, um peso para cada link que se conecta a outro switch e uma mensagem de roteamento que chega

Calcular:

uma tabela de roteamento atualizada

Método:

Mantenha um campo de *distância* em cada entrada da tabela de roteamento;
Inicialize a tabela de roteamento com uma entrada única que tem o
destino igual ao switch de pacote local, o *próximo hop* não-usado e a
distância configurada para zero;

Repita sempre {

espere a próxima mensagem de roteamento chegar através da rede a partir de
um vizinho; Faça com que o remetente seja o switch *N*;

para cada entrada na mensagem {

Faça *V* ser o destino na entrada e faça *D* ser a distância;

Calcule *C* como *D* mais o peso atribuído ao link através da
mensagem que chegou;

Examine e atualize a tabela de roteamento local;

se (não existe nenhuma rota para *V*) {

acrescente uma entrada para a tabela de roteamento local para o
destino *V* com próximo hop *N* e distância *C*;

} senão se (existe uma rota que tenha o próximo hop *N*) {

substitua a distância na rota existente com *C*;

} senão se (existe uma rota com distância maior que *C*) {

modifique o próximo hop para *N* e distância para *C*;

}

}

Algoritmo 13.2 Algoritmo de vetor de distâncias para cálculo de rota usada por cada switch de pacotes em uma WAN. Cada switch envia periodicamente a lista de pares (destino, distância) de sua tabela de roteamento para todos os seus vizinhos.

Como um algoritmo de vetor de distâncias, um algoritmo SPF pode se adaptar a falhas de hardware. Além disso, o SPF tem a vantagem de que todas as computações podem ser executadas simultaneamente – depois da mudança de status de um link, todos os switches de pacote recebem uma mensagem de status, e cada um começa a computar sua tabela de roteamento. Em contraste, um algoritmo por vetor de distâncias exige que um switch de pacotes atualize sua tabela de roteamento antes de enviar uma mensagem para outro switch de pacotes.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

Muitas tecnologias de WAN foram criadas, incluindo ARPANET, X.25, Frame Relay, SMDS e ATM. Atualmente, Frame Relay e SMDS fornecem serviços de WAN de alta velocidade, e muitas companhias telefônicas usam Frame Relay para proporcionar serviços de WAN de alta velocidade.

Exercícios

- 13.1 Se sua organização usa uma Rede de Longo Alcance, descubra a distância coberta pela rede e o número de computadores que ela liga.
- 13.2 A Figura 13.3 mostra como podem ser atribuídos endereços a computadores que se conectam a um switch de pacotes. Suponha que o hardware de uma das interfaces em um switch falhe, e um administrador da rede move a conexão de um computador para uma interface não-usada. A nova configuração trabalhará corretamente? Por que ou por que não?
- 13.3 Escreva um programa de computador que leia uma tabela de roteamento e então leia pacotes de quatro arquivos que simulem quatro interfaces de computador. Encaminhe cada pacote de acordo com a tabela de roteamento. Lembre-se de tratar pacotes que têm endereço incorreto.
- 13.4 O roteamento default ajuda a simplificar tabelas de roteamento. Por exemplo, suponha que uma WAN consiste em dois switches de pacote. Cada switch pode ter uma entrada de tabela de roteamento para cada endereço local (ou seja, o endereço de cada computador que está ligado ao switch) mais uma rota default que aponta para outro switch. Sob que circunstâncias o esquema falha?
- 13.5 Escreva um programa de computador que implemente o algoritmo de Dijkstra para encontrar caminhos mais curtos em um grafo.
- 13.6 Leia o Algoritmo 13.2. Quando os programas de computador executando em dois switches de pacote trocam informações de vetor de distâncias, os programas devem concordar com relação a um formato de mensagem. Projete um formato de mensagem que não seja ambíguo.
- 13.7 Estenda o exercício anterior implementando um programa de computador que use o formato de mensagem especificada. Faça com que outro estudante implemente um programa seguindo a mesma especificação e veja se eles interoperam corretamente.
- 13.8 Contate um provedor de rede comercial para descobrir quanto custa por ano alugar uma conexão de Frame Relay, uma conexão ATM e um circuito digital entre Nova York e São Francisco. Por que a diferença em custo é muito grande?
- 13.9 Uma tecnologia de rede pode ser usada para conexões de WAN e LAN? (Dica: leia o próximo capítulo.)
- 13.10 Descubra se seu site usa tecnologia ATM como uma WAN ou uma LAN.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

Se uma rede de telefonia digital usa pacotes largos, o atraso utilizado para preencher um pacote torna difícil o cancelamento de eco.

Para permitir que os comutadores de pacotes operem em altas velocidades e tenham baixo atraso, baixo jitter e cancelamento de eco, a tecnologia ATM divide todos os dados em pequenos pacotes de tamanho fixo, chamados de células. Cada célula ATM contém exatamente 53 bytes: 5 bytes de cabeçalho e 48 bytes de dados. A Figura 14.1 ilustra o formato de uma célula ATM.

Como mostra a figura, a maioria dos bits no cabeçalho são destinados a dois campos, rotulados VPI e VCI¹. Esses campos juntos identificam o destino da célula. Outros campos especificam um tipo de carga e dão um CRC de 8 bits para verificar que a célula não foi danificada durante o tráfego. O campo rotulado PRIO é um bit de Prioridade de Perda de Célula (*Cell Loss Priority*), que identifica se um pacote pode ser descartado quando a rede fica congestionada.

Lembre que o ATM foi projetado para ser completamente genérico e que o tamanho das células foi escolhido como ponto intermediário entre as células grandes, melhores para o transporte de dados, e as células pequenas, melhores para o transporte de voz. Os defensores argumentam que tal valor intermediário é necessário se uma única tecnologia tem de lidar com toda a comunicação possível. Os críticos sugerem que a escolha desse valor impede o ATM de otimizar tanto voz quanto dados.

Os críticos do ATM também criticam o tamanho relativo do cabeçalho, especialmente para transmissão de dados. Quando o ATM foi projetado, os projetistas resolveram limitar o cabeçalho a dez por cento do tamanho da carga. Por causa disso, uma carga de 48 bytes foi escolhida e o cabeçalho foi fixado em 5 bytes.

Para redes de dados, os dez por cento do ATM são um sobrepeso extremamente alto. Compare o custo da Ethernet, por exemplo, na qual pacotes podem ser de até 1500 bytes, com apenas 14 bytes de cabeçalho (ou seja, o cabeçalho tem somente um por cento do tamanho dos dados). Os engenheiros que criticam o sobrepeso do ATM se referem a ele como *cell tax*.

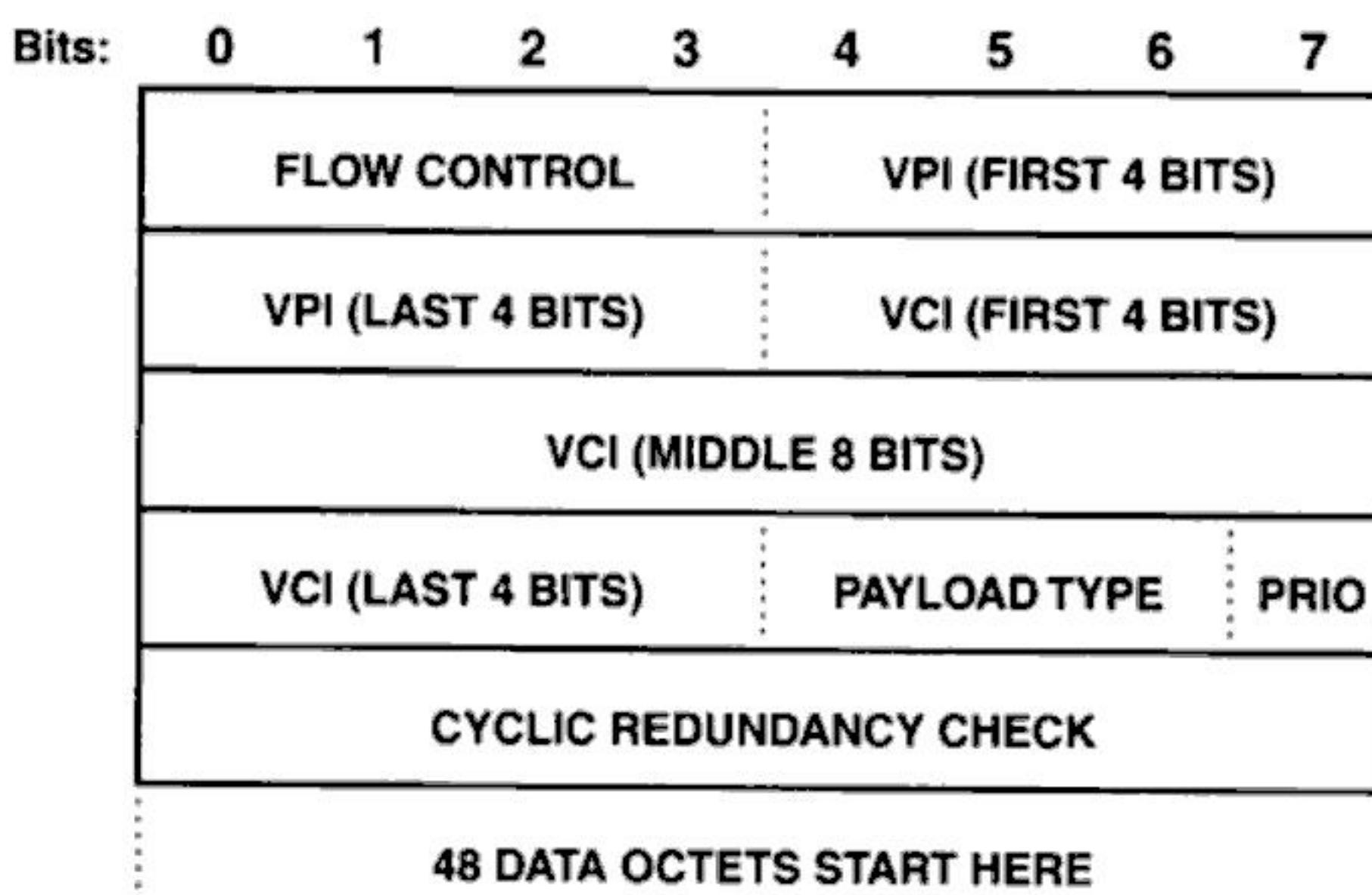


Figura 14.1 Campos nos cabeçalhos de 5-bytes encontrados numa célula de ATM. Cada linha no diagrama representa um byte.

¹ O formato mostrado é a Interface de Rede-Usuário (User Network Interface, UNI); o formato da Interface de Rede-Rede (Network Network Interface, NNI) tem um campo VPI levemente maior.

14.5 Serviço orientado à conexão

Assim como outras tecnologias de rede desenvolvidas pela indústria telefônica, o ATM usa um *paradigma de serviço orientado à conexão* (*connection-oriented service paradigm*). Antes que dois computadores possam se comunicar, eles precisam estabelecer uma "conexão" através da rede. Um dos computadores pede uma conexão ao outro (análogo a fazer uma discagem telefônica), que precisa aceitá-la (análogo a atender ao telefone). Depois que ambos concordam em se comunicar, o hardware de rede subjacente estabelece um caminho para os dados, chamado de *conexão*, e retorna um identificador de conexão (um valor binário) para os dois computadores.

Uma vez que a conexão tenha sido estabelecida, os dois computadores podem trocar dados. O computador transmissor gera uma seqüência de células, coloca o identificador de conexão em cada uma e passa o resultado para ser transportado pela rede. Quando recebe uma célula, o switch ATM extrai o identificador de conexão e consulta uma tabela para determinar como enviar a célula.

14.6 VPI/VCI

Formalmente, uma conexão ATM é conhecida como *Canal Virtual* (*Virtual Channel, VC*). O termo virtual é apropriado porque as conexões ATM são formadas por valores guardados na memória, em vez de por cabos físicos. O termo canal é menos descriptivo. Muitos profissionais expandem o acrônimo VC para o termo mais descriptivo *Círculo Virtual* (*Virtual Circuit*).

O ATM designa a cada VC um identificador de 24 bits, que é dividido em duas partes para produzir uma hierarquia. A primeira parte, o *Identificador de Caminho Virtual* (*Virtual Path Identifier, VPI*), especifica o caminho que o VC segue através da rede. A segunda parte, o *Identificador de Canal Virtual* (*Virtual Channel Identifier, VCI*), satisfaz um único VC dentro do caminho. Um VCI tem 16 bits de tamanho. Embora o hardware de rede possa usar a hierarquia para agrupar múltiplos canais virtuais, um computador usando ATM não interpreta os dois campos. Em vez disso, o computador vê as duas partes como um único valor binário de 24 bits que gera o identificador de conexão. Uma vez que o computador se refere às duas partes juntas, o identificador é conhecido como VPI/VCI.

14.7 Rótulos e comutação de rótulos

Uma rede ATM é formada por um ou mais dispositivos conhecidos como *switches ATM*. Cada switch ATM tem múltiplos pontos de ligação; um ponto de ligação pode conectar a um computador de usuário ou a outro switch. Informalmente, os pontos de ligação são conhecidos como *portas*. Por isso, a mais simples rede ATM consiste em um único switch ATM com múltiplos computadores conectados².

Surpreendentemente, um switch ATM muda o VPI/VCI em cada célula que gerencia. Dentro de cada switch há uma *tabela de encaminhamento* (*forwarding table*) que especifica como o hardware vai passar as células adiante. Cada entrada na tabela corresponde a um possível VPI/VCI para uma determinada *porta*; o switch usa o VPI/VCI em uma célula entrante para localizar a entrada correspondente na sua tabela de encaminhamento. Além do número de *porta* física para o qual a célula será enviada, a entrada da tabela contém um VPI/VCI substituto. O switch reescreve o VPI/VCI no cabeçalho da célula com o valor substituto, e encaminha a célula. Por isso, diferentemente dos endereços discutidos anteriormente, o VPI/VCI em uma célula não permanece o mesmo conforme uma célula atravessa a rede. Mudar o VPI/VCI é conhecido como *reescrever o rótulo* ou *comutar o rótulo* (*label rewriting* ou *label switching*), e o ATM é caracterizado como *sistema da comutação de rótulos* (*label switching system*).

Para entender as consequências de mudar o rótulo, considere uma rede ATM que consiste em um switch e dois computadores, como ilustra a Figura 14.2.

² Para uma ilustração, veja a Figura 8.11 na página 117.

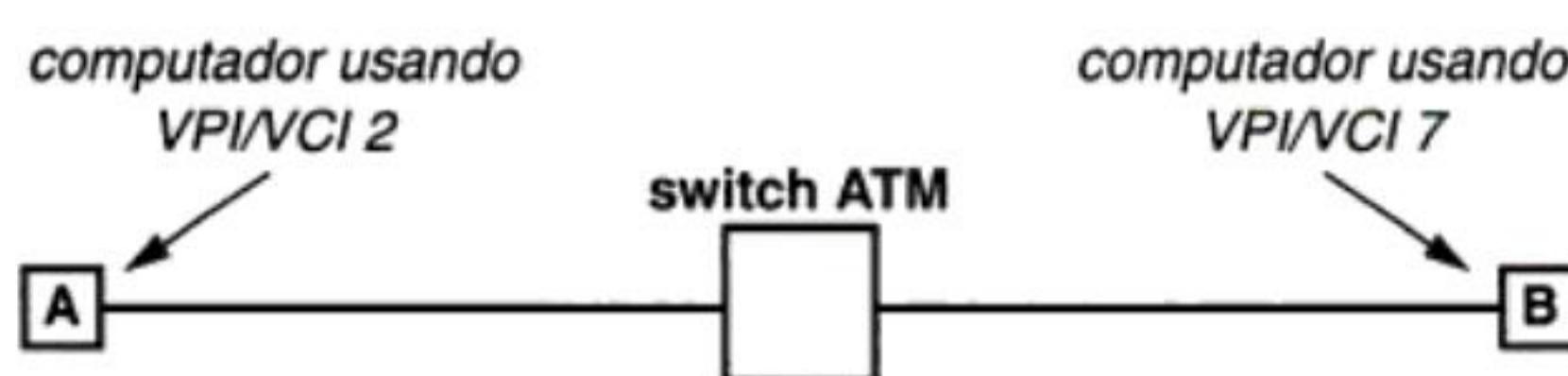


Figura 14.2 Ilustração do VC ATM entre dois computadores.

Na figura, os computadores A e B formaram um canal virtual entre eles. Quando o VC foi formado, o switch assinalou VPI/VCI 2 para o computador A e VPI/VCI 7 para o computador B. Quando designou os dois valores de VPI/VCI, o switch inicializou entradas na sua tabela de encaminhamento para mapeá-los. Depois, quando recebe uma célula do computador A com VPI/VCI 2, o switch muda o VPI/VCI para 7 antes de encaminhar a célula para o computador B, e vice-versa. Cada computador conhece o VPI/VCI que precisa utilizar para a conexão, mas nenhum sabe o VPI/VCI que o outro está usando. Para resumir:

Um switch ATM usa comutação de rótulos para reescrever o identificador de conexão (VPI/VCI) em cada célula que encaminha. Dois computadores usando um determinado VC geralmente têm valores diferentes para o VPI/VCI.

14.8 Um exemplo de viagem através da rede ATM

Considere a forma como a comutação de rótulos se estende num ATM de larga escala. As tabelas de encaminhamento de uma série de switches têm de ser coordenadas para criar um caminho através da rede. Por exemplo, a Figura 14.3 ilustra um conjunto de três switches ATM e mostra os valores nas tabelas de encaminhamento que correspondem a um único VC entre dois computadores.

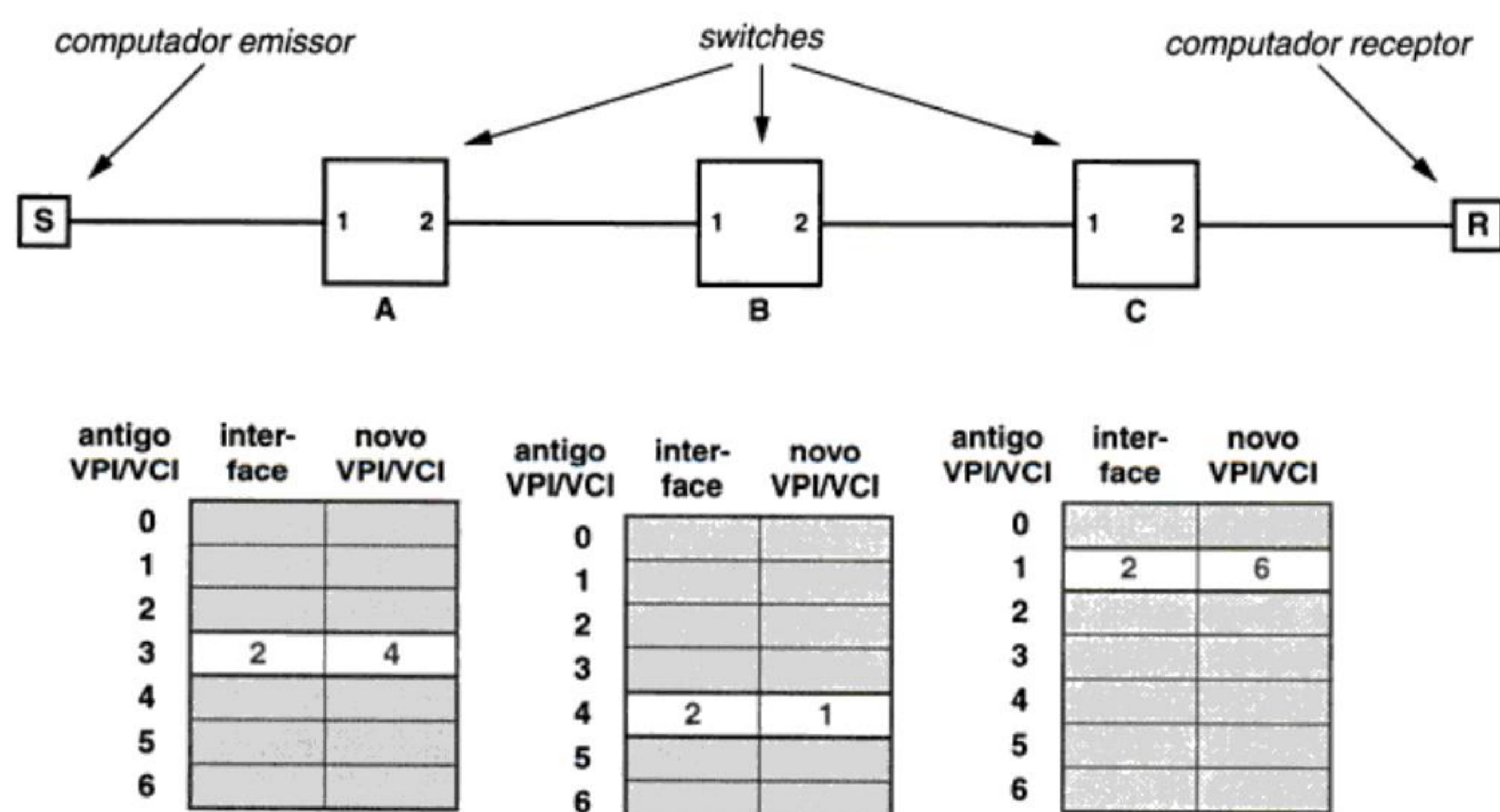


Figura 14.3 Uma ilustração de três switches ATM e as tabelas de encaminhamento de cada switch que permite a um único VC se propagar pela rede. Apenas as entradas das tabelas que correspondem ao VC são mostradas.

A figura mostra as entradas nas tabelas para um único VC do computador rotulado S (transmissor) ao computador rotulado R (receptor). Ao enviar e receber células através do VC, o computador S usa VPI/VCI 3 e o computador R usa VPI/VCI 6. Isto é, o computador S coloca 3 no cabeçalho de uma célula a ser enviada e então a transfere para o switch A. O switch A extrai o VPI/VCI da célula entrante e verifica a entrada numero 3 na sua tabela de encaminhamento. A tabela especifica que o VPI/VCI deve ser mudado para 4, o que A realiza antes de transmitir a célula pela interface 2 ao switch B. O switch B verifica o VPI/VCI da célula entrante e muda seu valor para 1 antes de encaminhá-la. O switch C verifica a entrada 1 em sua tabela e muda o valor para 6 antes de encaminhar a célula ao computador R. Como resultado, quando uma célula passa através do VC de S para R, ela toma os seguintes valores de VPI/VCI, em seqüência: 3, 4, 1 e 6. O sentido é:

Conforme uma célula passa através de uma rede ATM, o identificador muda a cada switch. As tabelas de encaminhamento em cada switch precisam estar coordenadas para definir caminhos significativos de um computador a outro através da rede.

14.9 Circuitos virtuais permanentes

Quando e como são preenchidas as entradas em uma tabela de encaminhamento ATM? Como são coordenadas as entradas através de um grande número de switches para garantir que estão corretas? Esta seção responde a essas perguntas.

Como o ATM foi projetado para fornecer todos os serviços de rede, o projeto precisa incluir mecanismos que correspondam às tecnologias de rede existentes. Por exemplo, para interconectar dois sites distantes, um cliente especifica exatamente quais dois pontos devem ser conectados, e então paga uma tarifa mensal para alugar um circuito digital de uma companhia telefônica. A companhia telefônica mantém o circuito no lugar enquanto o cliente pagar a tarifa. E um circuito digital sobrevive a quedas de computadores e mesmo a faltas de energia – o circuito pode ser utilizado novamente logo após a energia ter sido restaurada.

O ATM oferece um serviço análogo ao circuito digital alugado: um cliente pode requisitar que um provedor de rede ATM instale estabeleça um *Canal Virtual Permanente (Permanent Virtual Channel, PVC)*. O cliente especifica quais dois computadores devem ser interconectados e o fornecedor de rede estabelece um VC entre eles. A analogia entre um PVC de ATM e um circuito digital é forte. Como um circuito digital convencional, um PVC está pronto para ser usado a qualquer hora. Um PVC sobrevive a quedas de energia, bem como ao reinício dos computadores – um computador pode usar um PVC para transferir dados assim que a energia tiver sido restaurada.

Para estabelecer um PVC, o administrador de rede configura manualmente as entradas nas tabelas de encaminhamento. O termo técnico usado pelas companhias telefônicas (e pelo ATM) para tais configurações é provisioning; uma vez que um PVC foi estabelecido, diz-se que ele foi fornecido (provisioned).

Atualmente, provisioning requer dois passos. Primeiro, o administrador da rede determina o caminho completo (ou seja, a seqüência de switches ATM) de um computador à outro através da rede. Segundo, o administrador precisa escolher um VPI/VCI para ser usado a cada passo ao longo do caminho e configurar cada par de switches adjacentes de forma que o VPI/VCI saindo de um switch corresponda ao VPI/VCI entrando no próximo.

A habilidade do ATM de cambiar rótulos a cada switch torna o provisioning fácil, pois uma célula de ATM não precisa conter o mesmo VPI/VCI globalmente. Em vez disso, um administrador que configura uma PVC precisa considerar apenas um par de switches por vez. Para estender um PVC do switch S para o switch adjacente T, o administrador precisa apenas escolher um VPI/VCI que naquele momento não esteja em uso em T. Depois de reservar uma entrada na tabela T, o administrador pode preencher a entrada seguinte na tabela do switch S.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

períodos de comunicação intensa separados por longos minutos de silêncio. Para tais aplicativos, o ATM define uma *Taxa de Bits Disponíveis (Available Bit Rate, ABR)*. O tráfego numa conexão ABR pode usar toda a largura de banda disponível num dado momento.

14.12 A motivação para células e comutação de rótulos

Surgem diversas perguntas. Por que os projetistas do ATM escolheram células de tamanho fixo em vez de pacotes de tamanho variado? Qual é a maior vantagem da comutação de rótulos e por que ela foi adotada pelo ATM? As respostas são a mesma: para satisfazer taxas de transferência de dados máximas e os requerimentos QoS.

14.12.1 Células versus pacotes

A pesquisa nos anos 80 mostrou que o hardware de redes podia ser otimizado se todos os pacotes fossem exatamente do mesmo tamanho e tivessem exatamente o mesmo formato de cabeçalho. Primeiro, pacotes de tamanhos variáveis podem causar fragmentação da memória; células de tamanho fixo não. Segundo, pacotes de tamanhos variados requerem que o hardware acomode o maior tamanho possível para um pacote e detecte o final deste; o hardware pode simplesmente contar os bits de uma célula de tamanho fixo. Terceiro, uma vez que o tempo requerido para transmitir um pacote de tamanho variável depende de seu comprimento, o hardware de interface que transmite pacotes de tamanho variado tem de interagir com o switch. Usualmente o switch começa a transmissão e espera que a interface o interrompa. Finalmente, pacotes de tamanhos variados tornam difícil garantir a QoS, especialmente o baixo jitter. O problema surge porque, uma vez iniciada, a transmissão de um pacote não pode ser interrompida. Então, se um pequeno pacote chega a um canal que requer baixo jitter, a transmissão não pode ser interrompida. Por isso, se um pequeno pacote chega num canal que requer baixo jitter depois de ter iniciado a transmissão de um pacote grande, o pacote pequeno precisa esperar, violando o requerimento QoS.

A velocidade é o mais crucial no centro de uma rede; para realizar seu sonho, o ATM necessita de mais velocidade do que qualquer outra tecnologia de rede. Em particular, o ATM foi projetado para acomodar um número arbitrário de usuários através do mundo, cada um dos quais poderia querer pagar por um alto desempenho. Por isso, os designers escolheram células de tamanho fixo, e não pacotes de tamanhos variados, prezando a eficiência.

Para alcançar taxas de bits mais altas, o ATM foi projetado para trabalhar através da fibra óptica. Uma porta típica num switch ATM trabalha em uma velocidade OC-3 (155 Mbps), ou maior, com os switches maiores e mais caros capazes de trabalhar em velocidades muito mais altas. Por exemplo, existem switches ATM que podem lidar com velocidade de gigabits; são esperadas taxas de dados ainda mais altas.

14.12.2 Comutação de rótulo versus roteamento

A comutação de rótulos é outra tecnologia que aumenta a velocidade e a capacidade do hardware. O hardware tem pouco a ver com quando as células chegam: a extração do VPI/VCI e a checagem da tabela podem ser implementadas diretamente no hardware; nenhum CPU é necessário. Como resultado, a capacidade bruta de um switch ATM é geralmente muito alta. Por exemplo, um switch ATM típico, projetado como substituto para LANs, têm uma capacidade de desempenho agregada de 2.4 Gbps.

14.13 Transmissões de dados no ATM e no AAL5

Embora o serviço ATM descrito seja desenhado para aplicações como transmissões de voz telefônicas, aplicativos como os de transferência de dados foram otimizados para usar pacotes grandes. Para acomodar uma variedade de aplicativos, o ATM definiu um conjunto de protocolos de adapta-



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

media), o hardware não suporta broadcast ou multicast⁴. Em uma rede ATM o broadcast para um conjunto de computadores é simulado, arranjando um programa de aplicativo para passar uma cópia dos dados para cada computador no conjunto. Como resultado, broadcast é ineficiente.

Complexidade do QoS. Embora o padrão do ATM permita que um computador especifique os parâmetros QoS quando estabelece uma conexão, impor limites não é trivial, pois imposições fortes requerem que o hardware do switch compute médias estatísticas para cada conexão. Em particular, um switch não pode meramente computar o número médio de bits enviados numa unidade de tempo. Em vez disso, o ATM especifica que os dados podem ser enviados em pequenas explosões (bursts), com uma baixa taxa de dados e uma curta duração máxima dessas “explosões”. A complexidade das especificações torna o incremento pesado e difícil; muitas implementações não suportam o padrão completo. Mais importante, muitos especialistas argumentam que QoS de grãos finos não é necessário – ele não resolve problemas quando não há capacidade suficiente, e é desnecessário quando a capacidade é abundante.

Presunção de homogeneidade. O ATM é projetado para ser um sistema de rede único e universal. Há uma vontade mínima para interoperar com outras tecnologias. A lição que aprendemos a partir da Internet é que nenhuma tecnologia única basta; diferenças de custo e funcionalidade significam que existirão múltiplas tecnologias.

Por essas e outras razões, ATM ainda não se tornou a rede universal. Companhias telefônicas ainda usam-no em suas redes estruturais, e grandes corporações comumente têm uma ou mais redes ATM. Apenas algumas instituições tentaram usar o ATM como uma rede universal, e mesmo essas parecem estar reconsiderando sua escolha. Um dos impactos mais significativos no ATM veio quando engenheiros anunciaram que a transmissão de Gigabits pela Ethernet era viável, tornando possível a atualização de redes Ethernet existentes em companhias, em vez de substituí-las por tecnologias ATM mais caras. Outro impacto veio quando engenheiros anunciaram que foram bem-sucedidos em enviar o tráfego de protocolos de Internet (IP) diretamente através do SONET⁵. O ATM seguidamente usa o SONET como transporte subjacente, de forma que a habilidade de enviar o IP diretamente pelo SONET significa que os técnicos de células ATM podem ser eliminados.

14.15 Comutação de rótulos com múltiplos protocolos (Multiprotocol Label Switching, MPLS)

Embora o ATM não tenha se tornado popular, seus defensores continuam imaginando e promovendo variações de redes orientadas à conexão. Uma versão de rede orientada à conexão é chamada de Comutação de Rótulos com Múltiplos Protocolos (*Multiprotocol Label Switching, MPLS*). Em vez de tentar substituir completamente a Internet, MPLS foi projetada para servir no núcleo da Internet. A idéia básica é usar switches similares aos ATM para fornecer circuitos virtuais entre roteadores de IP⁶.

⁴ A especificação do ATM inclui um circuito virtual de ponto-a-multiponto para ajudar aplicativos a simular broadcast; um dos exercícios pede ao leitor para investigar o tópico mais a fundo.

⁵ O Capítulo 12 discute o SONET

⁶ O Capítulo 17 descreve roteadores de IP.

14.16 Resumo

As companhias telefônicas criaram a Rede Digital de Serviços Integrados (ISDN) e o Modo de Transferência Assíncrono (ATM). O ATM foi previsto como uma tecnologia universal de rede que lida com transmissões de voz, vídeo e dados. O ATM usa um paradigma orientado à conexão, no qual um aplicativo primeiramente cria um Canal Virtual (VC), usa tal canal para comunicação e então o encerra. A conexão é implementada por um ou mais switches ATM; cada um coloca uma entrada para o VC em sua tabela de encaminhamento.

Existem dois tipos de VCs ATM: um PVC é criado manualmente e sobrevive a quedas de força, e um SVC é criado sob pedidos. Quando cria um VC, um computador precisa especificar os requerimentos da Qualidade de Serviço (QoS); o hardware do ATM reserva o pedido de recursos ou recusa o pedido.

O ATM não tem sido aceito amplamente. Embora algumas companhias telefônicas ainda o utilizem em suas redes estruturais, o custo, a complexidade e a falta de interoperacionalidade impediram o ATM de prevalecer.

Exercícios

- 14.1** O bit PRIO no cabeçalho de uma célula especifica quando um switch ATM pode descartar a célula se o switch não puder guardar todas as células. Como pode um aplicativo usar o bit PRIO?
- 14.2** A separação do VPI e do VCI significa que o ATM pode usar uma hierarquia de encaminhamento. Descreva os detalhes do algoritmo que permite usar tal hierarquia.
- 14.3** Examine o padrão ATM. Qual é a diferença entre os serviços de Taxa de Bits Disponíveis (ABR) e Taxa de Bits Não Especificada (UBR)?
- 14.4** Leia mais sobre os VCs de ponto-a-multiponto da ATM. Se um conjunto de N aplicativos precisar participar em um grupo de multicast, quantos VCs de ponto-a-multiponto serão necessários?
- 14.5** Projete um esquema que permita a um conjunto de programas de aplicativos N participar de um multicast em uma rede ATM sem usar ponto-a-multiponto. Quantos VCs precisam ser estabelecidos?
- 14.6** Leia sobre emulação de LANs (LANE), um esforço para fornecer a conectividade da Ethernet em uma infraestrutura de ATM. A LANE é sensível? Por quê?
- 14.7** Considere a implementação de um aplicativo de acesso a uma base de dados remota. Se você pudesse escolher entre usar ATM ou uma tecnologia de LAN convencional como a Ethernet, qual você preferiria? Por quê?
- 14.8** Existe algum aplicativo de dados que funcionaria melhor no ATM do que em uma rede de pacotes convencionais? Caso positivo, descreva-os e comente por que. Se não, explique por que não.
- 14.9** Se você pudesse substituir toda a Internet por uma rede ATM, você o faria? Por quê?



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

Camada 4: Transporte

Os protocolos da Camada 4, que especificam como tratar dos detalhes de transferência confiável, estão entre os mais complexos. Este capítulo discute os problemas de transporte básico e os capítulos posteriores mostram um exemplo de um protocolo de transporte.

Camada 5: Sessão

Os protocolos da Camada 5 especificam como estabelecer uma sessão de comunicação com um sistema remoto (por exemplo, como fazer um login em um computador remoto de tempo compartilhado). As especificações para detalhes de segurança, como autenticação, usando senhas pertencem à Camada 5.

Camada 6: Apresentação

Os protocolos da Camada 6 especificam como representar dados. Tais protocolos são necessários porque diferentes marcas de computadores usam representações internas diferentes para inteiros e caracteres. Deste modo, os protocolos da Camada 6 são necessários para traduzir da representação em um computador para a representação em outro.

Camada 7: Aplicativo

Cada protocolo da Camada 7 especifica como um aplicativo em particular usa uma rede. Por exemplo, a especificação para um aplicativo que transfere arquivos de um computador para outro pertence à Camada 7. O protocolo especifica os detalhes de como um programa aplicativo em uma máquina faz um pedido (por exemplo, como especificar o nome do arquivo desejado) e como o aplicativo em outra máquina responde.

16.6 Pilhas: divisão de software em camadas

Quando são projetados protocolos de acordo com um modelo de camadas, o software de protocolo resultante segue a organização em camadas. O software de protocolo em cada computador é dividido em módulos, com um módulo correspondendo a cada camada. O mais importante é que a divisão em camadas determina as interações entre os módulos: em teoria, quando o software de protocolo envia ou recebe dados, cada módulo se comunica somente com o módulo correspondente da camada imediatamente superior ou inferior. Assim, os dados que estão sendo enviados passam para baixo através de cada camada, e os dados em recebimento passam para cima através de cada camada. A Figura 16.2 ilustra o conceito.

Como mostra a figura, cada computador contém software para todo um conjunto de protocolos. Os vendedores usam o termo *pilha (stack)* para se referir a tal software porque o modelo em camadas usado na construção do software é freqüentemente ilustrado por um conjunto de retângulos, como nas Figuras 16.1 e 16.2. Deste modo, a pergunta “Qual pilha o seu computador está executando?” normalmente se refere a protocolos de rede, e não a uma estrutura de dados de pilha.

Nos últimos trinta anos, muitos representantes comerciais criaram pilhas de protocolos. Atualmente, contudo, a maioria das pilhas foi substituída por protocolos TCP/IP. A tabela na Figura 16.3 lista cinco pilhas de protocolos históricas.

Como as pilhas têm sido projetadas independentemente, os protocolos de uma determinada pilha não podem interagir com protocolos de outras. Desta forma, se o dono de um computador escolher usar a pilha de *AppleTalk* da Apple Computer Corporation em um computador, este computador poderá se comunicar somente com outros computadores que usem a pilha de *AppleTalk*.

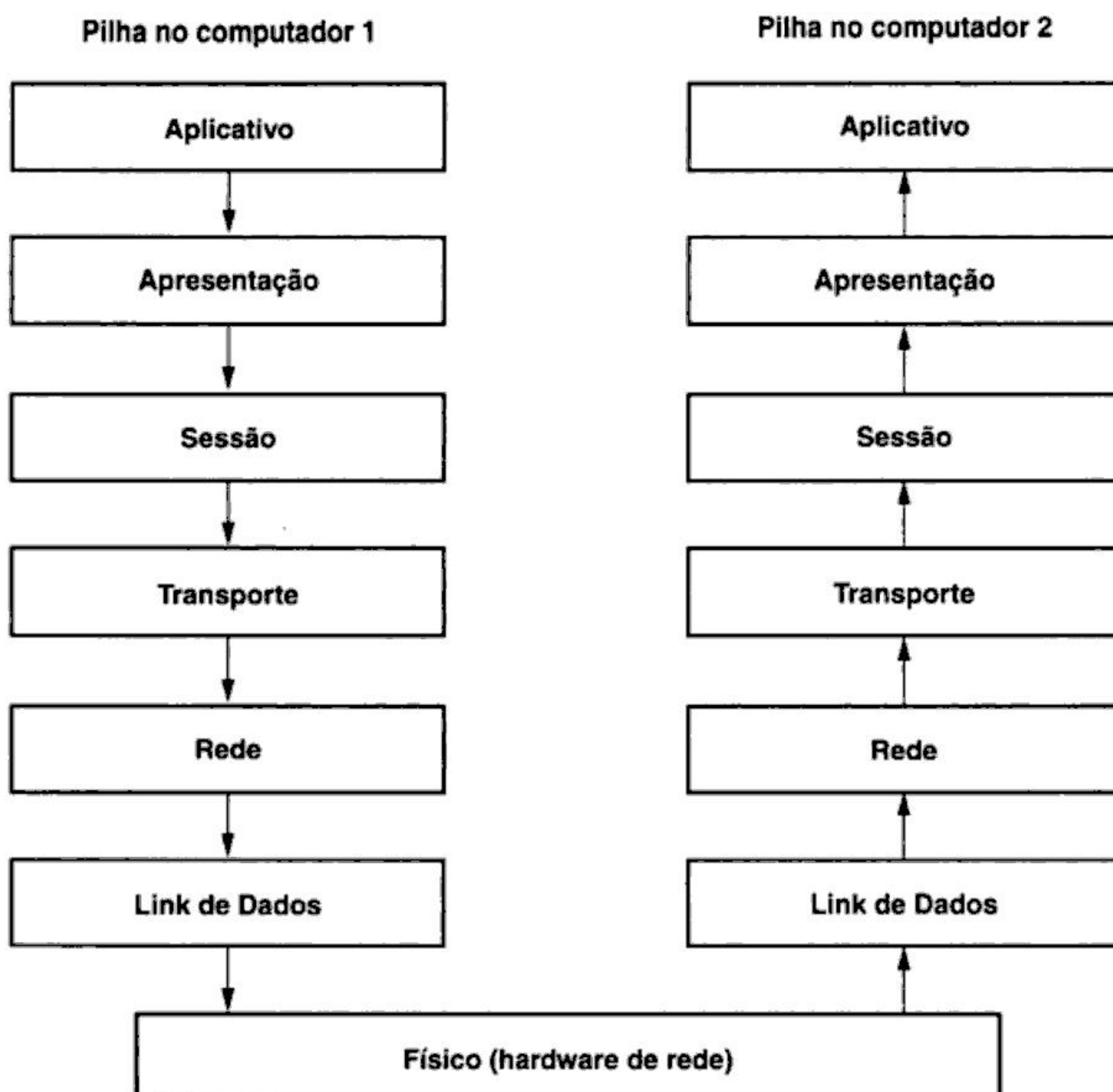


Figura 16.2 O caminho conceitual de dados conforme viajam de um aplicativo no computador 1 através de uma rede para um aplicativo no computador 2.

Se necessário, um computador pode executar mais de uma pilha ao mesmo tempo. Por exemplo, um computador pode ter software para uma pilha de AppleTalk e também para uma pilha de TCP/IP. Duas pilhas podem rodar no mesmo computador e transmitir através de uma rede física única sem interferência, porque o campo de tipo em cada quadro identifica que pilha deve tratar a mensagem. Isto é, se um computador tem duas pilhas, a informação do tipo nos quadros que chegam especifica qual pilha deve processar o quadro.

Vendedor	Pilha
Novell Corporation	Netware
Banyan System Corporation	VINES
Apple Computer Corporation	AppleTalk
Digital Equipment Corporation	DECNET
IBM (muitos vendedores)	SNA

Figura 16.3 Exemplos de pilhas de protocolo. Embora as pilhas compartilhem muitos conceitos gerais, os detalhes diferem, tornando-as incompatíveis.

16.7 Como o software em camadas funciona

Foi dito que cada camada de software de protocolo resolve uma parte do problema de comunicação. Para fazer isso, o software em uma determinada camada no computador remetente acrescenta informações aos dados que estão sendo enviados, e o software na mesma camada no computador receptor usa as informações adicionais para processar os dados recebidos. Por exemplo, suponha que tenha sido criada uma pilha de protocolo na qual a camada de *Link de Dados* tenha recebido a responsabilidade pelos checksums. A camada de *Link de Dados* no computador transmissor tem que computar o checksum, e a camada de *Link de Dados* no computador receptor tem que verificar se o checksum está correto. Na Figura 16.2, sempre que um quadro de partida chega ao software de *Link de Dados* no computador 1, o software computa um checksum antes de enviar o quadro através da rede. Sempre que um quadro entrante chega ao software de *Link de Dados* no computador 2, o software verifica e remove o checksum antes de passar o quadro à camada de *Rede*.

16.8 Múltiplos cabeçalhos aninhados

Normalmente, cada camada coloca informações adicionais em um cabeçalho antes de enviar dados a uma camada inferior. Deste modo, um quadro que está viajando através de uma rede contém uma série de cabeçalhos aninhados, como mostra a Figura 16.4.

Como mostra a figura, o cabeçalho correspondente ao protocolo de nível mais baixo ocorre primeiro. Na prática, o cabeçalho para a segunda camada, o protocolo de *Link de Dados*, ocorre primeiro – embora a Camada 1 especifique os sinais elétricos ou ópticos usados para transmitir um quadro, ela não acrescenta um cabeçalho da mesma forma que outras camadas o fazem.

A Figura 16.4 ilustra um conceito geral, mas não mostra todas as possibilidades. Em particular, alguns softwares de protocolo fazem mais do que adicionar um cabeçalho no início de dados de partida. Por exemplo, o Capítulo 6 mostra que um protocolo de link de dados pode acrescentar no início um caractere especial para marcar o começo de um quadro, acrescentar outro caractere especial para marcar o fim e inserir caracteres adicionais no meio para esconder ocorrências de caracteres especiais. Semelhantemente, alguns protocolos especificam que todas as informações adicionais devem ser acrescentadas no final dos quadros em um *trailer*, em vez de em um *cabeçalho* no início dos mesmos.

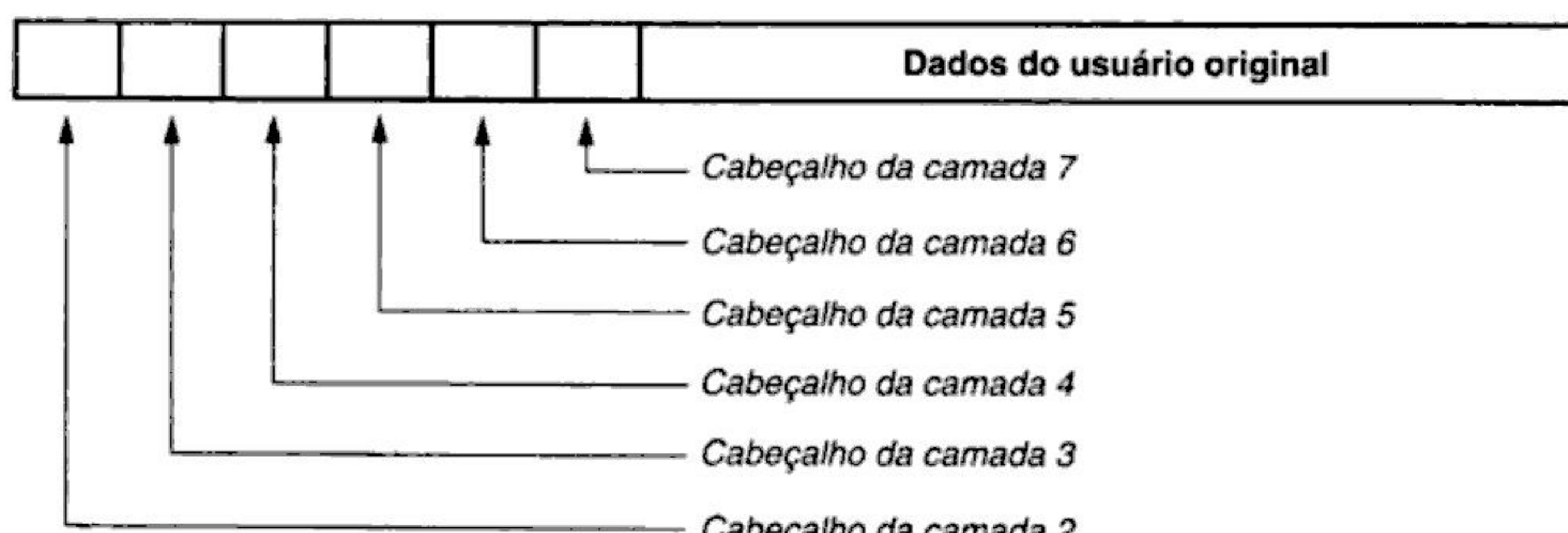


Figura 16.4 Os cabeçalhos de protocolo aninhados que aparecem em um quadro à medida que este viaja através de uma rede se toda a pilha ISO for usada. Cada camada de software de protocolo acrescenta um cabeçalho para um quadro que está sendo enviado.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

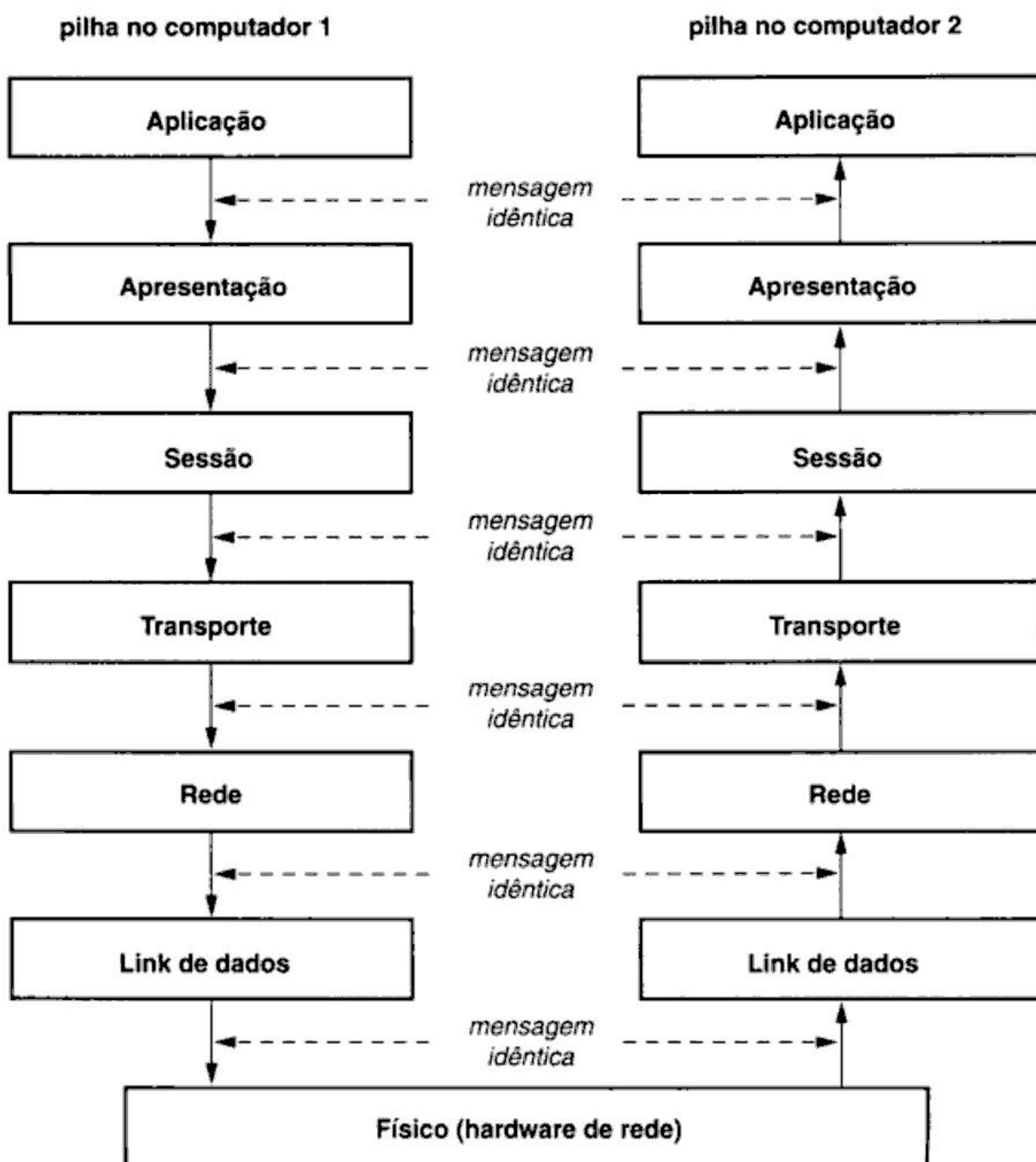


Figura 16.5 O princípio da divisão em camadas aplicado a cada camada do antigo modelo ISO. Se o software de protocolo no computador remetente muda a mensagem, a mudança deve ser revertida pelo software de protocolo correspondente no receptor.

sua lista para ver se podem ser entregues também pacotes adicionais. Se o pacote chegou fora de ordem, o software de protocolo acrescenta-o à lista.

16.10.2 Seqüenciamento para eliminar pacotes duplicados

Um hardware com mau funcionamento pode fazer com que os pacotes sejam duplicados. A duplicação surge freqüentemente em WANs, mas pode acontecer também em LANs. Por exemplo, um transceiver com mau funcionamento em uma LAN que usa CSMA/CD pode levar o receptor a perceber uma transmissão válida enquanto o remetente percebe uma colisão. Como resultado da colisão, o remetente fará uma espera (back off) e retransmitirá, fazendo com que duas cópias de um quadro cheguem ao receptor³.

³ O autor passou por essa situação em primeira mão.

O seqüenciamento resolve o problema da duplicação. O software receptor verifica se há duplicatas quando examina o número de seqüência de um pacote que está sendo recebido. Se o pacote já foi entregue ou o número da seqüência for igual ao de um dos pacotes que estão esperando na lista, o software descarta a nova cópia.

16.10.3 Retransmitindo pacotes perdidos

A perda de pacotes é um problema fundamental em redes de computadores, porque erros de transmissão podem adulterar bits, tornando o quadro inválido. Quando um receptor detecta tais problemas, ele descarta o quadro.

Para garantir a transferência (isto é, transferência sem perda), os protocolos usam *confirmação positiva com retransmissão*. Sempre que um quadro chega intacto, o software de protocolo receptor devolve uma pequena mensagem que relata que a recepção foi bem-sucedida. A mensagem é conhecida como confirmação (*acknowledgment, ACK*). O remetente tem a responsabilidade de assegurar que cada pacote seja transferido com sucesso. Sempre que o remetente envia um pacote, o software de protocolo do lado remetente começa um temporizador. Se uma confirmação chega antes do temporizador expirar, o software cancela tal temporizador. Se o temporizador expira antes de uma confirmação chegar, o software envia outra cópia do pacote e começa o temporizador novamente. A ação de enviar uma segunda cópia é conhecida como *retransmitir (retransmitting)* e a cópia é comumente chamada de *retransmissão (retransmission)*.

A retransmissão não pode ter sucesso se um defeito de hardware desconectou permanentemente a rede ou se o computador receptor sofreu um *crash*. Portanto, os protocolos que retransmitem mensagens usualmente limitam o número máximo de retransmissões. Quando o limite é alcançado, o protocolo pára de retransmitir e declara que é impossível se comunicar.

Note que a retransmissão pode introduzir pacotes duplicados. Como um remetente não pode distinguir entre um pacote perdido e um pacote que sofreu um longo atraso, ele pode decidir retransmitir muito cedo. Eventualmente, é possível que ambas as cópias do pacote sejam entregues. Desta modo, os protocolos que usam retransmissões para fornecer confiabilidade devem tratar também do problema de pacotes duplicados.

16.10.4 Evitando repetições causadas por atraso excessivo

Uma fonte de atraso em um sistema de comutação de pacotes advém da abordagem de *armazenamento e encaminhamento*. Quando um pacote chega em um switch de pacote, ele é colocado em uma fila. Se os pacotes chegam mais rápido do que o switch pode encaminhá-los, a fila de pacotes será grande e o atraso pode ser excessivo. Atrasos extraordinários podem levar a *erros de replay (replay errors)*. *Replay* significa que um pacote antigo que esteja atrasado afeta uma comunicação posterior. Por exemplo, considere a seguinte seqüência de eventos.

- Dois computadores concordam em se comunicar às 13h.
- Um computador envia uma seqüência de dez pacotes para o outro.
- Um problema de hardware faz com que o pacote 3 sofra um atraso.
- Algumas rotas mudam para evitar o problema de hardware.
- O software de protocolo no computador remetente retransmite o pacote 3, e os pacotes restantes são transmitidos sem erro.
- Às 13h 05min, os dois computadores concordam em se comunicar novamente.
- Depois de o segundo pacote chegar, a cópia atrasada do pacote 3, pertencente à conversação antiga, chega.
- O pacote 3 da segunda conversação chega.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

17.3 O conceito de serviço universal

O principal problema com múltiplas redes deveria ser óbvio: um computador acoplado a uma determinada rede pode se comunicar somente com outros computadores acoplados à mesma rede. O problema se tornou evidente nos anos 70 quando grandes organizações começaram a adquirir múltiplas redes. Cada rede na organização formou uma ilha. Em muitas instalações iniciais, cada computador estava acoplado a uma única rede e os funcionários tinham que escolher um computador apropriado para cada tarefa. Isto é, um funcionário recebia acesso a múltiplas telas e teclados, e era forçado a se mover de um computador a outro para enviar uma mensagem através da rede apropriada.

Os usuários não ficam satisfeitos nem são produtivos quando têm que usar um computador separado para cada rede. Conseqüentemente, a maioria dos sistemas de comunicação modernos permite a comunicação entre quaisquer dois computadores de maneira análoga a um sistema telefônico que fornece comunicação entre quaisquer dois telefones. Conhecido como *serviço universal*, o conceito é uma parte fundamental da área de redes¹. Com serviço universal, um usuário em qualquer computador em qualquer parte de uma organização pode enviar mensagens ou dados para qualquer outro usuário. Além disso, um usuário não precisa mudar de sistema de computador quando muda tarefas – todas as informações estão disponíveis para todos os computadores. Como resultado, os usuários são mais produtivos. Para resumir: Um sistema de comunicação que fornece serviço universal permite que pares arbitrários de computadores se comuniquem. O serviço universal é desejável porque aumenta a produtividade individual.

Um sistema de comunicação que fornece serviço universal permite que pares arbitrários de computadores se comuniquem. O serviço universal é desejável porque aumenta a produtividade individual.

17.4 Serviço universal em um mundo heterogêneo

O serviço universal significa que uma organização precisa adotar uma única tecnologia de rede, ou é possível ter serviço universal através de múltiplas redes que usam múltiplas tecnologias? No Capítulo 11, aprendemos que incompatibilidades elétricas impossibilitam a formação de uma grande rede apenas se interconectando os fios de duas redes. Além disso, técnicas de extensão como ligação com bridges (*bridging*) não podem ser usadas com tecnologias de rede heterogêneas porque tecnologias diferentes usam formatos de pacote e esquemas de endereçamento incompatíveis. Deste modo, um quadro criado para uma tecnologia de rede não pode ser transmitido em uma rede que usa uma tecnologia diferente. A questão pode ser resumida:

Embora serviço universal seja altamente desejável, incompatibilidades entre hardwares de rede e endereçamentos físicos evitam que uma organização construa uma rede com bridges que inclua tecnologias arbitrárias.

17.5 Ligação inter-redes

Apesar das incompatibilidades entre tecnologias de rede, os pesquisadores inventaram um esquema que fornece serviço universal entre redes heterogêneas. O esquema, chamado *ligação inter-redes (internetworking)*, utiliza hardware e software. Os sistemas de hardware adicionais são usados para interconectar um conjunto de redes físicas. O software em todos os computa-

¹ Muitas tecnologias tentaram fornecer serviço universal: o *Modo de Transferência Assíncrono (Asynchronous Transfer Mode, ATM)* é um exemplo recente.

dores acoplados fornece serviço universal. O sistema que resulta das redes físicas conectadas é conhecido como uma *inter-rede (internetwork)*.

A ligação inter-redes é bastante geral. Em particular, uma inter-rede não é restrita em tamanho – existem inter-redes que contêm poucas redes e outras que contêm milhares de redes. Semelhantemente, o número de computadores acoplados a cada rede em uma inter-rede pode variar – algumas redes não têm nenhum computador acoplado, enquanto outras têm centenas.

17.6 Conexão de rede física com roteadores

O componente básico de hardware usado para conectar redes heterogêneas é o *roteador*. Fisicamente, os roteadores se assemelham a bridges – cada roteador é um sistema de hardware de propósito específico dedicado à tarefa de interconectar redes. Como uma bridge, um roteador tem processador e memória convencionais, bem como uma interface de E/S separada para cada rede que conecta. A rede trata uma conexão com um roteador da mesma forma que uma conexão com qualquer outro computador. A Figura 17.1 mostra que a conexão física de redes com um roteador é direta.

A figura usa uma nuvem para representar cada rede, em vez de uma linha ou um círculo, porque as conexões de roteadores não são restritas a uma determinada tecnologia de rede. Um roteador pode conectar duas LANs, uma LAN e uma WAN ou duas WANs. Além disso, quando um roteador conectar duas redes na mesma categoria geral, as redes não precisam usar a mesma tecnologia. Por exemplo, um roteador pode conectar uma LAN Ethernet com uma rede de Frame Relay. Deste modo, cada nuvem representa uma tecnologia de rede arbitrária.

Para resumir:

Um roteador é um computador de propósito especial dedicado à tarefa de interconectar redes. Um roteador pode interconectar redes que usam tecnologias diferentes, incluindo meio, esquema de endereçamento físico ou formato de quadro diferentes.

17.7 Arquitetura de inter-rede

Os roteadores possibilitam a uma organização escolher tecnologias de rede apropriadas para cada necessidade e usar roteadores para conectar todas as redes em uma única inter-rede. Por exemplo, a Figura 17.2 mostra como podem ser usados três roteadores para conectar quatro redes físicas arbitrárias em uma inter-rede.

Embora a figura mostre cada roteador com exatamente duas conexões, os roteadores comerciais podem conectar mais de duas redes. Deste modo, um único roteador poderia conectar todas as quatro redes em nosso exemplo. Porém, uma organização raramente usa um único roteador para conectar todas as suas redes. Existem duas razões:

- Como o roteador deve encaminhar cada pacote, o processador em um determinado roteador não é suficiente para lidar com o tráfego que passa por um número arbitrário de redes.

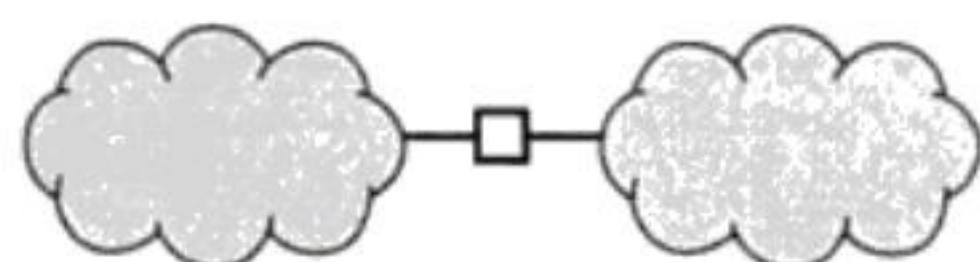


Figura 17.1 Duas redes físicas conectadas por um roteador, que tem uma interface separada para cada conexão de rede. Os computadores podem ser acoplados a cada rede.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

18.8 Classes e notação decimal pontilhada

A notação decimal pontilhada trabalhava bem com endereços IP de classes porque o IP usa limites de octeto para separar um endereço em prefixo e sufixo. Em um endereço classe A, os últimos três octetos correspondem a um sufixo de host. Semelhantemente, os endereços classe B têm dois octetos de sufixo de host, e os de classe C têm um octeto.

Infelizmente, como a notação decimal pontilhada não mostra os bits individuais de um endereço, a classe deve ser reconhecida a partir do valor decimal do primeiro octeto. A Figura 18.4 mostra o alcance decimal de valores para cada classe.

18.9 Divisão do espaço de endereçamento

O esquema de classe IP não divide o espaço de endereçamento de 32 bits em classes de tamanho igual, e as classes não contêm o mesmo número de redes. Por exemplo, a metade de todos os endereços IP (aqueles em que o primeiro bit é zero) pertence à classe A. Surpreendentemente, a classe A pode conter somente 128 redes, porque o primeiro bit de um endereço classe A deve ser zero e o prefixo ocupa um octeto. Deste modo, apenas sete bits permanecem disponíveis para numerar as redes classe A. A Figura 18.5 resume o número máximo de redes disponíveis em cada classe e o número máximo de hosts por rede.

Como mostra a figura, o número de bits alocados a um prefixo ou sufixo determina quantos números únicos podem ser atribuídos. Por exemplo, um prefixo de n bits permite 2^n números de rede únicos, enquanto um sufixo de n bits permite que 2^n números de host sejam atribuídos a uma determinada rede.

Classe	Faixa de valores
A	0 a 127
B	128 a 191
C	192 a 223
D	224 a 239
E	240 a 255

Figura 18.4 O alcance decimal de valores encontrados no primeiro octeto de cada classe de endereço.

Classe do endereço	Bits no prefixo	Número máximo de redes	Bits no sufixo	Número máximo de hosts por rede
A	7	128	24	16777216
B	14	16384	16	65536
C	21	2097152	8	256

Figura 18.5 O número de redes e hosts por rede em cada uma das três classes primárias de endereço IP.

⁵ Para ajudar uma organização a escolher endereços, o RFC 1597 recomenda endereços de classes A, B e C, que podem ser usados em inter-redes privadas.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

18.17.4 O endereço deste computador

Um computador precisa saber seu endereço IP para enviar ou receber pacotes de inter-rede, porque cada pacote contém o endereço da origem e do destino. O suíte de protocolos TCP/IP contém os protocolos que um computador pode usar para obter seu endereço IP automaticamente quando o computador realiza o *boot*. Curiosamente, os protocolos de inicialização (*startup*) usam IP para se comunicar. Ao usar tais protocolos de inicialização, um computador não pode fornecer um endereço IP de origem correto. Para tratar de tais casos, o IP reserva o endereço que consiste em todos os bits zero para significar *este computador*.

18.17.5 Endereço de loopback

O IP define um *endereço de loopback* usado para testar aplicativos de rede. Os programadores freqüentemente usam o teste de loopback para a depuração preliminar depois de ser criado um aplicativo de rede. Para executar um teste de loopback, um programador deve ter dois programas aplicativos que planejam se comunicar através de uma rede. Cada aplicativo inclui o código necessário para interagir com o software de protocolo TCP/IP. Em vez de executar cada programa em um computador separado, o programador executa os dois programas em um único computador e os instrui a usar um endereço de loopback IP ao se comunicar. Quando um aplicativo envia dados para outro, os dados descem a pilha de protocolos até o software IP, que os encaminha para cima através da pilha de protocolos até o segundo programa. Deste modo, o programador pode testar rapidamente a lógica do programa sem precisar de dois computadores e sem enviar pacotes através de uma rede.

O IP reserva o prefixo de rede classe A 127 para usar com loopback. O endereço de host usado com 127 é irrelevante – todos os endereços de host são tratados da mesma forma. Por convenção, os programadores usam freqüentemente o número de host 1, tornando 127.0.0.1 a forma mais popular de loopback.

Durante o teste de loopback, nenhum pacote deixa um computador – o software IP encaminha pacotes de um programa aplicativo a outro. Conseqüentemente, o endereço de loopback nunca aparece em um pacote que viaja através de uma rede.

18.18 Resumo de endereços IP especiais

A tabela na Figura 18.8 resume as formas de endereço IP especiais.

Dissemos que os endereços especiais são reservados e que nunca devem ser atribuídos a hosts. Além disso, cada endereço especial é restrito a certos usos. Por exemplo, um endereço de broadcast nunca deve aparecer como um endereço de origem, e o endereço com todos Os não deve ser usado depois de um host completar o procedimento de inicialização e obter um endereço IP.

Prefixo	Sufixo	Tipo de endereço	Propósito
todos Os rede	todos Os	deste computador rede	usado durante a inicialização identifica uma rede
rede	todos 1s	broadcast direcionado	em uma rede específica
todos 1s	todos 1s	broadcast limitado	em uma rede local
127	qualquer	de loopback	teste

Figura 18.8 Resumo das formas de endereço IP especiais.

18.19 A forma de endereço de broadcast de Berkeley

A University of California, em Berkeley, desenvolveu e distribuiu uma primeira implementação de protocolos TCP/IP como parte do UNIX BSD⁹. A implementação do BSD continha uma característica não-padronizada que afetou muitas implementações subsequentes. Em vez de usar um sufixo de host com todos os bits em um para representar um endereço de broadcast direcionado, a implementação de Berkeley usou um sufixo de host que continha tudo em zero. A forma de endereço é informalmente conhecida como *broadcast de Berkeley*.

Infelizmente, muitos fabricantes de computadores derivaram suas primeiras versões do software TCP/IP a partir da implementação de Berkeley, e alguns poucos sites ainda usam broadcast de Berkeley. Algumas implementações TCP/IP incluem um parâmetro de configuração que permite escolher entre o TCP/IP padrão e a forma de Berkeley; muitas implementações são construídas para aceitar as formas de endereço broadcast padrão e de Berkeley.

18.20 Roteadores e o princípio de endereçamento IP

Além de designar um endereço de inter-rede para cada host, o Protocolo de Internet especifica que os roteadores deveriam ter endereços IP atribuídos também. Cada roteador é designado com dois ou mais endereços IP. Para entender por quê, recorde dois fatos:

- Um roteador tem conexões com múltiplas redes físicas.
- Cada endereço IP contém um prefixo que especifica uma rede física.

Deste modo, um único endereço IP não basta para um roteador porque cada roteador se conecta a múltiplas redes. O esquema IP pode ser explicado por um princípio fundamental:

Um endereço IP não identifica um computador específico. Em vez disso, cada endereço IP identifica uma conexão entre um computador e uma rede. Em um computador com múltiplas conexões de rede (por exemplo, um roteador) precisa ser atribuído um endereço IP para cada conexão.

A Figura 18.9 ilustra a idéia com um exemplo que mostra endereços IP atribuídos a dois roteadores que conectam três redes.

O IP não exige que o mesmo sufixo seja atribuído a todas as interfaces de um roteador. Na figura, por exemplo, o roteador conectando a Ethernet e a rede Wi-fi tem sufixos 99.5 (conexão com a Ethernet) e 2 (conexão com o Wi-fi). Porém, o IP não previne o uso do mesmo sufixo em todas as conexões. Deste modo, o exemplo mostra que o administrador escolheu usar o mesmo sufixo, 17, para ambas as interfaces do roteador que conectam a rede Wi-fi com a WAN. Por uma questão prática, usar o mesmo sufixo pode ajudar os administradores da inter-rede porque é mais fácil lembrar um único número.

18.21 Hosts multi-homed

Um host pode ter múltiplas conexões de rede? Sim. Um computador de host que conecta múltiplas redes é chamado de *multi-homed*. Homing múltiplo é às vezes usado para aumentar a confiabilidade – se uma rede falha, o host pode ainda alcançar a inter-rede através da segunda co-

⁹ BSD significa *Berkeley Software Distribution*.

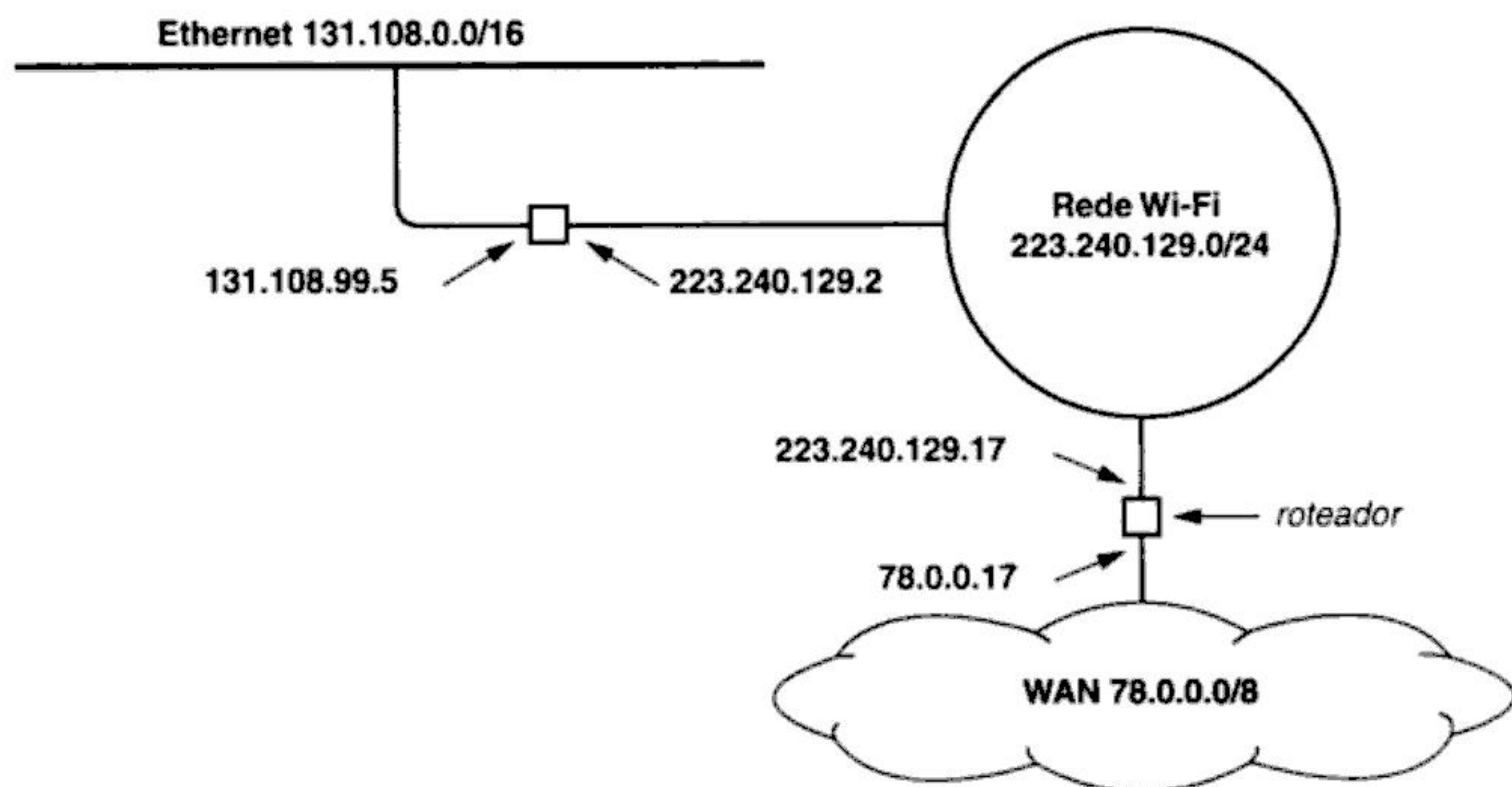


Figura 18.9 Um exemplo de endereços IP atribuídos a dois roteadores. A cada interface é atribuído um endereço que contém o prefixo da rede à qual a interface se conecta.

nexão. Alternativamente, homing múltiplo é usado para aumentar o desempenho – conexões com múltiplas redes possibilitam o envio de tráfego diretamente e podem evitar roteadores, que estão às vezes congestionados. Como um roteador, um host multi-homed tem múltiplos endereços de protocolo, um para cada conexão de rede.

18.22 Resumo

Para fornecer a ilusão de uma grande rede integrada, uma inter-rede usa um esquema de endereçamento uniforme. A cada computador é atribuído um endereço de protocolo; usuários, programas aplicativos e a maioria dos protocolos usam o endereço de protocolo quando estão se comunicando.

No TCP/IP, o Protocolo de Internet (*Internet Protocol*, IP) especifica o endereçamento. O IP divide cada endereço de inter-rede em uma hierarquia de dois níveis: o prefixo de um endereço identifica a rede a qual o computador se liga, e o sufixo identifica um computador específico na rede. Para assegurar que os endereços permaneçam únicos ao longo de uma determinada inter-rede, uma autoridade central deve atribuir prefixos de rede. Uma vez que um prefixo foi atribuído, um administrador de rede local pode atribuir a cada host na rede com um sufixo único.

Um endereço IP é um número de 32 bits. Originalmente, o endereço pertencia a uma de cinco classes, em que a classe de um endereço era determinada pelo valor dos primeiros quatro bits. A uma rede física que contivesse entre 257 e 65.536 hosts era atribuído um prefixo de classe *B*; a redes menores era atribuído um prefixo de classe *C*, e cada uma das redes maiores era designada com um prefixo de classe *A*.

O esquema de endereços IP original foi estendido para permitir que a divisão entre prefixo e sufixo ocorresse em um limite de bits arbitrário. Para fazê-lo, os endereçamentos sem classes e de sub-rede (CIDR) armazenam uma máscara de 32 bits junto com cada endereço. A máscara tem valor de 1 para cada bit no prefixo, e valor de zero para cada bit no sufixo.

Embora seja conveniente pensar em um endereço IP como se especificasse um computador, cada endereço IP identifica uma conexão entre um computador e uma rede. Roteadores e hosts multi-homed, que têm conexões com múltiplas redes físicas, devem ter múltiplos endereços IP.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

de amarração de endereços na Figura 19.2 corresponde a uma rede com o prefixo $197.15.3.0/24$. Portanto, cada endereço IP na tabela começará com o prefixo de 24 bits $197.15.3$. As implementações podem economizar espaço omitindo o prefixo das entradas da tabela.

A principal vantagem da abordagem de pesquisa de tabela é a generalidade – uma tabela pode armazenar as amarrações de endereço para um conjunto arbitrário de computadores em uma determinada rede. Em particular, um endereço de protocolo pode ser mapeado para um endereço de hardware arbitrário. Além disso, o algoritmo de pesquisa de tabela para resolução de endereços é direto e está entre os mais fáceis de programar. Dado um endereço IP do próximo hop, N , o software varre a tabela até encontrar uma entrada em que o endereço IP seja igual a N . O software então extrai o endereço de hardware da entrada.

Para uma rede que contenha menos de uma dúzia de hosts, uma pesquisa seqüencial pode bastar – o software de resolução começa na primeira entrada e procura seqüencialmente em cada uma até que uma igualdade seja encontrada. Para redes grandes, porém, uma pesquisa seqüencial exige tempo excessivo de UCP. Em tais casos, o software pode usar uma de duas implementações-padrão para melhorar a eficiência computacional: *hashing* ou indexação direta.

O *hashing* é uma técnica de estrutura de dados de propósito geral, bem conhecida para a maioria dos programadores. A indexação direta é ligeiramente mais eficiente, mas é uma técnica menos geral. A indexação direta somente é possível em casos em que os endereços de protocolo são designados a partir de um faixa compacta de valores². Por exemplo, a indexação direta pode ser usada com os endereços IP na Figura 19.2 porque os endereços são valores seqüenciais que começam com $197.15.3.2$. Para usar indexação direta com tais valores, o software mantém um array unidimensional de endereços de hardware e usa o sufixo de host de um endereço IP como um índice no array. A Figura 19.3 mostra a técnica.

A figura mostra como o mapeamento direto é usado para traduzir o endereço IP $197.15.3.5$. O software extrai o sufixo de host 5 e usa-o como um índice no array para obter o endereço de hardware $0A:74:59:32:CC:1F$. Na prática, para prevenir que um endereço IP ilegal cause um erro subscripto, o software deve verificar para se assegurar que o sufixo está dentro do alcance.

19.6 Resolução de endereço com computação de forma fechada

Recorde que, embora muitas tecnologias de rede usem endereços físicos estáticos, algumas usam endereçamento configurável, no qual uma interface de rede pode ser designada com um endere-

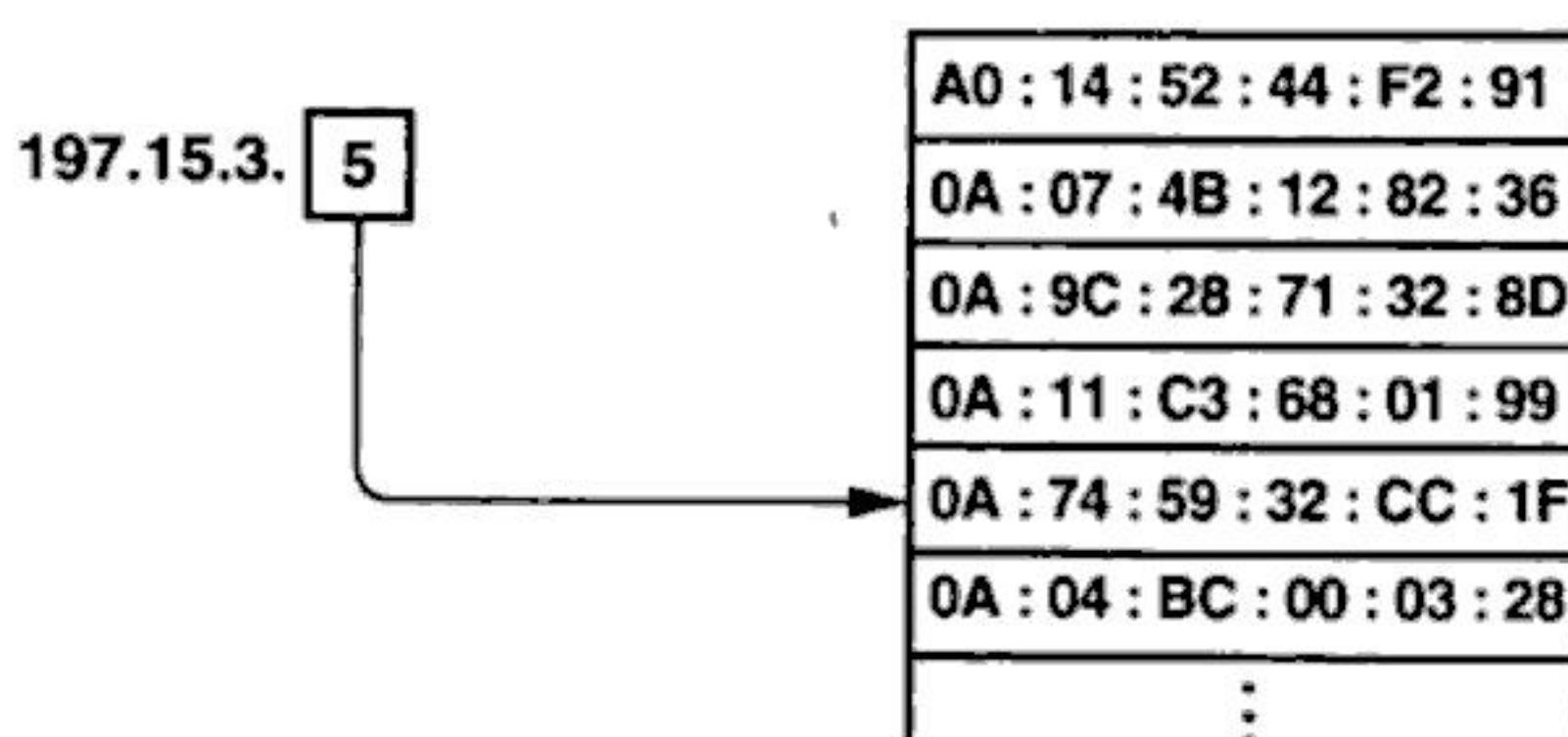


Figura 19.3 Um exemplo de pesquisa direta para uma rede de classe C. A porção de host de um endereço é usada como índice de array.

² Alguns administradores escolhem números não seqüenciais para endereços IP para ajudar a identificar o propósito de um computador (por exemplo, hosts são designados com sufixos abaixo de 200, enquanto roteadores são designados com sufixos acima de 200).

ço de hardware específico. Para tais redes, é possível escolher endereços que possibilitem a resolução de endereços de forma fechada. Um *resolver* que usa um método de forma fechada computa uma função matemática que mapeia um endereço IP para um endereço de hardware. Se a relação entre um endereço IP e o endereço de hardware correspondente é direta, a computação exige somente algumas operações aritméticas.

Para entender por que a computação de forma fechada pode ser especialmente eficiente para uma rede com endereços configuráveis, lembre que hardware e endereços IP podem ser mudados. Desse modo, valores podem ser escolhidos para otimizar a tradução. A porção de host do endereço IP de um computador pode ser escolhida para ser idêntica ao endereço de hardware do computador, tornando a tradução trivial.

Como um exemplo, suponha que a uma rede configurável foi atribuído o número de rede de $220.123.5.0/24$. À medida que computadores são acrescentados à rede, cada computador é designado com um sufixo de endereço IP e um endereço de hardware correspondente. O primeiro host é designado com o endereço IP $220.123.5.1$ e endereço de hardware 1. O segundo host é designado com o endereço IP $220.123.5.2$ e endereço de hardware 2. Os sufixos não precisam ser seqüenciais: se um roteador acoplado à rede é designado com o endereço IP $220.123.5.101$, o roteador é designado com o endereço de hardware 101. Dado o endereço IP de qualquer computador na rede, o endereço de hardware do computador pode ser computado por uma única operação Booleana and:

```
endereço_de.hardware = endereço_ip & 0xff
```

Deveria ser óbvio a partir do exemplo por que a resolução de forma fechada é freqüentemente usada com redes configuráveis. A computação é fácil de ser programada, não exige que uma tabela de valores seja mantida e é computacionalmente eficiente.

Para resumir:

Quando um computador se conecta a uma rede que usa endereçamento configurável, o administrador de rede local deve escolher um endereço de hardware, bem como um endereço IP. Os dois valores podem ser escolhidos para tornar a resolução de endereços trivial.

19.7 Resolução de endereços com troca de mensagem

Os mecanismos de resolução de endereços descritos podem ser computados usando um único computador em cada passo: as instruções e os dados necessários para a computação são mantidos no sistema operacional do computador. A alternativa para a computação local é uma abordagem distribuída, em que um computador que precisa resolver um endereço envia uma mensagem através de uma rede e recebe uma resposta. A mensagem carrega uma requisição que especifica o endereço de protocolo, e a resposta carrega o endereço de hardware correspondente.

Onde uma requisição de resolução de endereço deveria ser enviada? A maioria dos sistemas de protocolo escolhe entre dois projetos possíveis. No primeiro, uma rede inclui um ou mais servidores³ aos quais é designada a tarefa de responder requisições de resolução de endereços. Sempre que a resolução de um endereço for necessária, uma mensagem deve ser enviada para um dos servidores, que enviarão uma resposta. Em alguns suítes de protocolo, cada computador recebe o endereço de um ou mais servidores que podem ser usados – o computador envia uma mensagem para cada um deles em seqüência até que encontre um servidor ativo e receba uma resposta. Em outros

³ Capítulos posteriores descrevem servidores em detalhe e fornecem exemplos. Por enquanto, é suficiente encarar um servidor como um programa de computador capaz de se comunicar através de uma rede.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

19.9 Entrega de mensagem ARP

O padrão ARP especifica exatamente como mensagens ARP devem ser enviadas através de uma rede. Em particular, o protocolo especifica que uma mensagem de requisição ARP deve ser colocada em um quadro de hardware e difundida com broadcast para todos os computadores na rede. Cada computador recebe a requisição e examina o endereço IP. O computador mencionado na requisição envia uma resposta; todo os demais processam e descartam a requisição sem enviar uma resposta.

Quando um computador envia uma resposta ARP, a resposta não é enviada com broadcast, mas colocada em um quadro e enviada diretamente de volta ao computador que emitiu a requisição. A Figura 19.5 mostra uma troca ARP em computadores em uma Ethernet.

A figura mostra que, embora a mensagem de requisição ARP alcance todos os computadores, a resposta não. Veremos que o protocolo fornece as informações na requisição de transmissão pública e que todos os computadores as recebem quando do processamento da requisição.

19.10 Formato de mensagem ARP

Embora o Protocolo de Resolução de Endereço contenha uma especificação exata do formato de mensagem ARP, o padrão não fornece um formato fixo que deve ser usado para toda comunicação. O padrão ARP descreve a forma geral para mensagens ARP e especifica como determinar os detalhes para cada tipo de hardware de rede. A motivação para adaptar mensagens de ARP ao hardware existe porque uma mensagem ARP contém campos de endereços de hardware. Os projetistas do ARP perceberam que não podiam escolher um tamanho fixo para campos de endereço de hardware porque novas tecnologias de rede que tivessem endereços maiores do que o tamanho escolhido poderiam ser inventadas. Consequentemente, os projetistas incluíram um campo de tamanho fixo no princípio de uma mensagem ARP para especificar o tama-

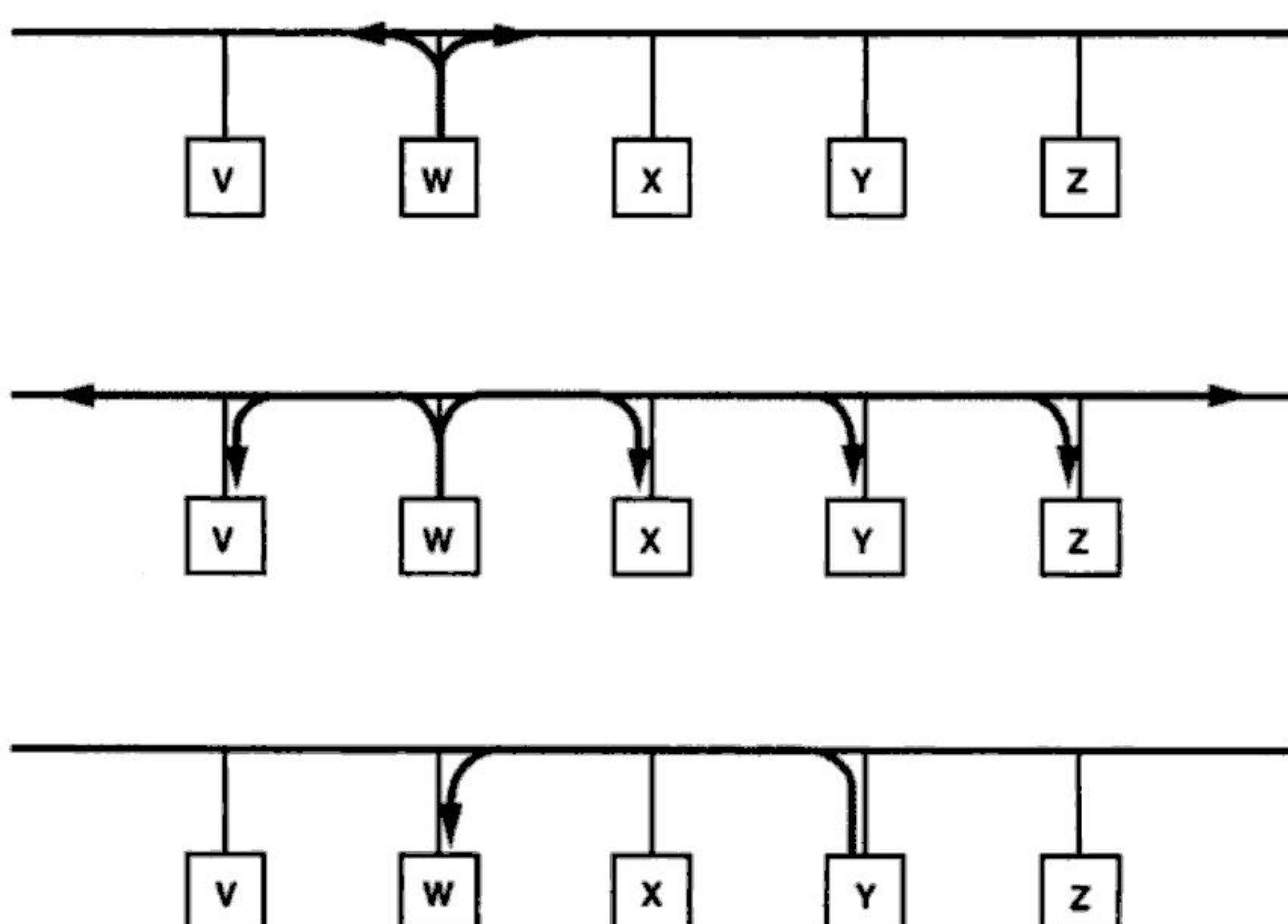


Figura 19.5 Uma troca de mensagens ARP. (a) O computador *W* inicia a difusão por broadcast de uma requisição ARP que contém o endereço IP do computador *Y*. (b) Todos os computadores recebem a requisição, e (c) o computador *Y* envia uma resposta diretamente para *W*.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

xima versão oficial do IP fosse 5. Porém, o número de versão 5 já tinha sido designado a um protocolo experimental conhecido como ST. Conseqüentemente, a nova versão do IP recebeu 6 como seu número de versão oficial, e o protocolo se tornou conhecido como *IPv6*¹.

22.5 Características do IPv6

O IPv6 retém muitas das características de projeto que fez o IPv4 ser tão bem sucedido. Como o IPv4, o IPv6 é sem conexão – cada datagrama contém um endereço de destino e é roteado independentemente. Como o IPv4, o cabeçalho em um datagrama contém um número máximo de hops que o datagrama pode passar antes de ser descartado. O IPv6 retém a maioria das facilidades gerais fornecidas pelas opções do IPv4.

Apesar de reter os conceitos básicos da versão corrente, o IPv6 modifica todos os detalhes. Por exemplo, ele usa endereços maiores e um formato completamente novo de cabeçalho de datagrama. O IPv6 usa também uma série de cabeçalhos de comprimento fixo para tratar de informações do cabeçalho. Então, ao contrário do Ipv4, que coloca informações-chave em campos fixos do cabeçalho e apenas utiliza apêndices de tamanhos variáveis para informações menos importantes, o cabeçalho do Ipv6 é sempre de tamanho variável.

As novas características do IPv6 podem ser agrupadas em cinco categorias principais:

- *Tamanho de endereço*. Em vez de 32 bits, cada endereço do IPv6 contém 128 bits. O espaço de endereçamento resultante é grande suficiente para acomodar o crescimento contínuo da Internet global por muitas décadas.
- *Formato de cabeçalho*. O cabeçalho de datagrama do IPv6 é completamente diferente do cabeçalho do IPv4. Quase todos os campos no cabeçalho foram mudados e alguns foram substituídos.
- *Cabeçalhos de extensão*. Diferente do IPv4, que usa um único formato de cabeçalho para todos os datagramas, o IPv6 codifica informações em cabeçalhos separados. Um datagrama consiste no cabeçalho base do IPv6 seguido por zero ou mais cabeçalhos de extensão, seguidos por dados.
- *Suporte para áudio e vídeo*. O IPv6 inclui um mecanismo que permite a um remetente e a um receptor estabelecer um caminho de alta qualidade através da rede subjacente e associar datagramas com aquele caminho. Embora o mecanismo vise ao uso com aplicativos de áudio e vídeo que exigem garantias de alto desempenho, o mecanismo pode ser usado também para associar com datagramas caminhos de baixo custo.
- *Protocolo extensível*. Diferente do IPv4, o IPv6 não especifica todas as características de protocolo possíveis. Em vez disso, os projetistas forneceram um esquema que permite a um remetente acrescentar informações adicionais a um datagrama. O esquema de extensão torna o IPv6 mais flexível do que IPv4, e permite que novas características sejam acrescentadas ao projeto quando necessário.

22.6 Formato de datagrama do IPv6

Como a Figura 22.1 mostra, um datagrama do IPv6 começa com um *cabeçalho de base*, seguido por zero ou mais *cabeçalhos de extensão*, seguidos por dados.

Embora ilustre a estrutura de datagrama geral, os campos na figura não são desenhados em escala. Em particular, alguns cabeçalhos de extensão são maiores do que o cabeçalho de base, enquanto outros podem ser pequenos. Além disso, em muitos datagramas o tamanho da área de dados é muito maior do que o dos cabeçalhos.

¹ Para distinguir o protocolo IP corrente do da nova versão, o corrente é chamado de *IPv4*.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

campo *TIME TO LIVE*), um roteador deve recalcular o checksum antes de encaminhar o datagrama para seu próximo hop¹.

A ação tomada em resposta a um erro de checksum é direta: o datagrama deve ser imediatamente descartado sem qualquer processamento. O receptor não pode confiar em quaisquer campos no cabeçalho do datagrama porque não tem como saber quais bits foram alterados. Em particular, o receptor não pode devolver uma mensagem de erro para o computador que enviou o datagrama porque não pode confiar no endereço de origem do cabeçalho. Igualmente, o receptor não pode encaminhar o datagrama danificado porque não pode confiar no endereço de destino do cabeçalho. Deste modo, o receptor não tem outra opção senão descartar o datagrama danificado.

23.3 Internet control message protocol (protocolo de mensagens de controle inter-rede)

Os problemas menos severos do que erros de transmissão resultam em condições de erro que podem ser relatadas. Por exemplo, suponha que alguns dos caminhos físicos em uma inter-rede falhem, fazendo com que ela seja particionada em dois conjuntos distintos de redes sem um caminho entre eles. Um datagrama enviado de um host em um conjunto para um host em outro não pode ser entregue.

O suíte TCP/IP inclui um protocolo que o IP usa para enviar mensagens de erro quando condições como a descrita acima surgem: o *Internet Control Message Protocol* (ICMP). O protocolo é necessário para uma implementação padrão de IP. Veremos que os dois protocolos são co-dependentes. O IP usa o ICMP quando envia uma mensagem de erro, e o ICMP usa o IP para transportar suas mensagens.

A Figura 23.1 lista as mensagens ICMP, que inclui mensagens de erro e mensagens informativas. Depois de revisar alguns exemplos, veremos como elas podem ser usadas.

Exemplos de mensagens de erros de ICMP incluem:

- *Source quench*. Um roteador envia um source quench sempre que recebe tantos datagramas que não tem mais espaço de buffer disponível. Um roteador que fica temporariamente sem espaço de buffer é obrigado a descartar datagramas que chegam. Quando descarta um datagrama, ele envia uma mensagem source quench para o host que gerou tal datagrama. Quando um host recebe um source quench, ele deve reduzir a taxa em que está transmitindo.
- *Time exceeded*. Uma mensagem time exceeded é enviada em dois casos. Sempre que um roteador decrementa para zero o campo *TIME TO LIVE* em um datagrama, ele descarta o datagrama e envia uma mensagem time exceeded. Além disso, uma mensagem time exceeded é enviada para um host se o temporizador (timer) de remontagem expira antes que todos os fragmentos de um determinado datagrama tenham chegado.
- *Destination unreachable*. Sempre que um roteador determina que um datagrama não pode ser entregue ao seu destino final, ele envia uma mensagem destination unreachable para o host que gerou tal datagrama. A mensagem especifica se é o host de destino específico ou a rede a qual está acoplado o destino que está incontactável (unreachable). Em outras palavras, a mensagem de erro distingue entre uma situação em que uma rede inteira está temporariamente desconectada de uma inter-rede (por exemplo, porque um roteador falhou) e o caso em que um host em particular está temporariamente fora do ar (por exemplo, por estar desligado).
- *Redirect*. Quando um host cria um datagrama destinado a uma rede remota, ele envia o datagrama a um roteador, que o encaminha para seu destino. Se um roteador determina que um host

¹ Um exercício sugere que, em vez de recalcular o checksum inteiro, um roteador possa obter maior desempenho mudando incrementalmente o checksum se a única mudança no cabeçalho for um decremento de TTL.

Tipo	Nome
0	Echo Reply
1	Não atribuído
2	Não atribuído
3	Destination Unreachable
4	Source Quench
5	Redirect
6	Alternate Host Address
7	Não atribuído
8	Echo
9	Router Advertisement
10	Router Selection
11	Time Exceeded
12	Parameter Problem
13	Timestamp
14	Timestamp Reply
15	Information Request
16	Information Reply
17	Address Mask Request
18	Address Mask Reply
19	Reservado (por Segurança)
20-29	Reservado (por Experimento de Força)
30	Traceroute
31	Datagram Conversion Error
32	Mobile Host Redirect
33	IPv6 Where-Are-You
34	IPv6 I-Am-Here
35	Mobile Registration Request
36	Mobile Registration Reply
37	Domain Name Request
38	Domain Name Reply
39	SKIP
40	Photuris
41-255	Reservado

Figura 23.1 Uma lista de mensagens ICMP. Cada uma é identificada por um campo *type* de 8 bits.

enviou incorretamente um datagrama, pois este deveria ter sido enviado para um roteador diferente, o roteador usa a mensagem redirect para fazer com que o host mude sua rota. A mensagem redirect pode especificar uma mudança para um host específico ou uma mudança para uma rede; a segunda opção é mais comum.

- *Parameter Problem*. Um dos parâmetros especificados em um datagrama está incorreto.

Além das mensagens de erro, o ICMP define mensagens informativas, que incluem:

- *Echo Request/Reply*. Uma mensagem echo request pode ser enviada para o software de ICMP em qualquer computador. Em resposta a uma mensagem echo request recebida, o software de ICMP deve enviar uma mensagem ICMP echo reply. A resposta carrega os mesmos dados que a requisição.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

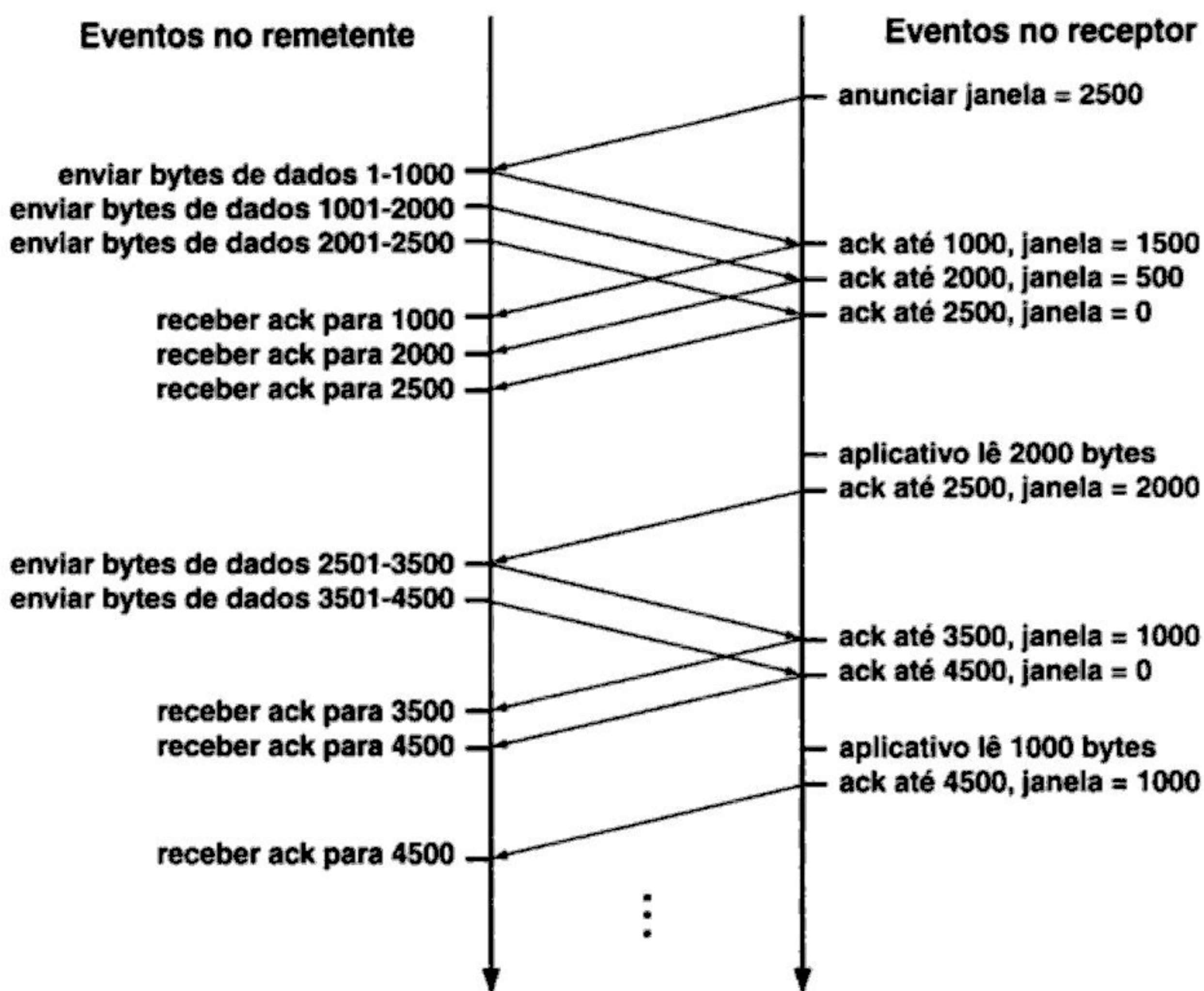


Figura 25.4 Uma seqüência de mensagens que mostra o controle de fluxo do TCP quando o tamanho de segmento máximo for de 1000 bytes. Um remetente pode transmitir dados suficientes para encher a janela atualmente anunciada.

tamanho de janela é sempre medido além dos dados que estão sendo confirmados, de forma que o receptor está anunciando que ele pode aceitar 2000 bytes além dos 2500 que já recebeu. O remetente responde com a transmissão de dois segmentos adicionais. Com a chegada de cada segmento, o receptor envia um acknowledgment com o tamanho de janela reduzido em 1000 bytes (isto é, a quantidade de dados que chegaram).

Uma vez mais, a janela alcança zero, fazendo com que o remetente pare a transmissão. Com o tempo, o aplicativo receptor consome alguns dados, e o TCP receptor transmite um acknowledgment com um tamanho de janela positivo. Se o remetente tiver mais dados a serem enviados, ele pode transmitir outro segmento.

25.11 Three-way handshake

Para garantir que conexões sejam estabelecidas ou terminadas de maneira confiável, o TCP usa um *3-way handshake*, em que três mensagens são trocadas¹. Cientistas provaram que a troca de 3 mensagens é necessária e suficiente para assegurar o acordo não ambíguo apesar da perda, duplicação e atraso de pacotes.

¹ Durante o *3-way handshake*, cada lado da conexão inclui um anúncio inicial de janela na mensagem que leva o SYN.

O TCP usa o termo *segmento de sincronização* (*segmento de SYN*) para descrever mensagens em um 3-way handshake usado para criar uma conexão, e o termo *segmento de FIN* (abreviação de *finish segment*) para descrever mensagens em um 3-way handshake usado para fechar uma conexão. A Figura 25.5 mostra o 3-way handshake usado para fechar uma conexão.

Como outras mensagens, o TCP retransmite segmentos de SYN ou de FIN perdidos. Além disso, o handshake garante que o TCP não abrirá ou fechará uma conexão até que ambos os lados tenham interagido.

Parte do 3-way handshake usado para criar uma conexão exige que cada lado gere aleatoriamente um número de seqüência de 32 bits. Se um aplicativo tenta estabelecer uma nova conexão de TCP após o *reboot* de um computador, o TCP escolhe um novo número aleatório. Porque cada nova conexão recebe uma nova seqüência aleatória, um par de programas aplicativos pode usar o TCP para se comunicar, fechar a conexão e então estabelecer uma nova conexão sem interferência de pacotes duplicados ou atrasados.

25.12 Controle de congestionamento

Um dos aspectos mais interessantes do TCP é um mecanismo para *controle de congestionamento*. Na maioria das inter-redes modernas, é mais provável que a perda de um pacote (ou um atraso extremamente longo) seja causada por congestionamento do que um defeito de hardware. Curiosamente, protocolos de transporte que retransmitem podem exacerbar o problema de congestionamento injetando cópias adicionais de uma mensagem. Se o congestionamento ativa retransmissão excessiva, o sistema inteiro pode atingir um estado de *colapso por congestionamento*, análogo a um engarrafamento em uma estrada. Para evitar o problema, o TCP sempre usa a perda de pacotes como uma medida de congestionamento e responde a esse congestionamento reduzindo a taxa em que retransmite dados.

Sempre que uma mensagem é perdida, o TCP inicia o controle de congestionamento. Em vez de retransmitir dados suficientes para encher o buffer do receptor (isto é, o tamanho da janela do receptor), o TCP inicia enviando uma única mensagem contendo dados. Se o acknowledgment chega sem perda adicional, o TCP dobra a quantidade de dados enviados e envia duas mensagens adicionais. Se acknowledgments chegam para aqueles dois, o TCP envia mais quatro, e assim por diante. O aumento exponencial continua até que o TCP esteja enviando metade da janela anunciada do receptor, quando o TCP diminui a taxa de aumento. O TCP então aumenta o tamanho da janela linearmente (enquanto o congestionamento não ocorrer).

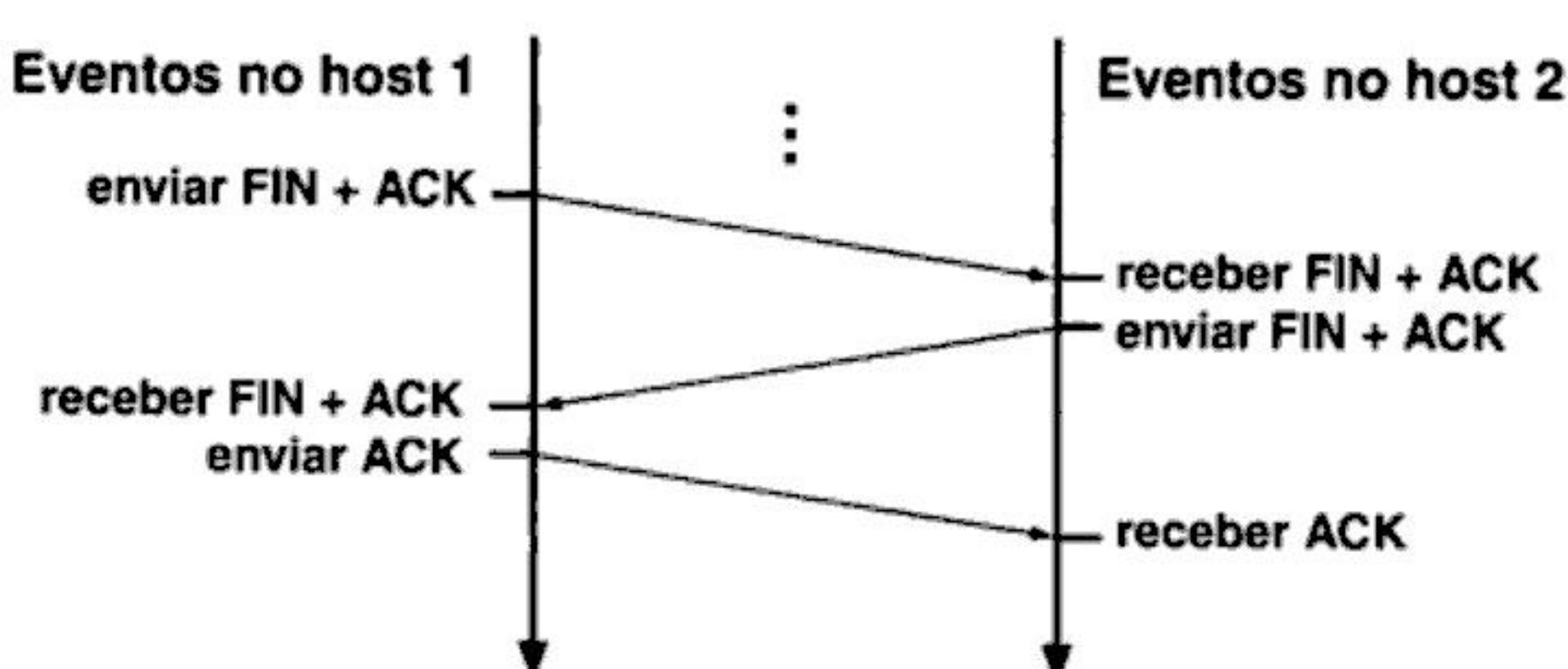


Figura 25.5 O 3-way handshake usado para fechar uma conexão. Os acknowledgments enviados em cada direção são usados para garantir que todos os dados chegaram antes da conexão ser terminada.

O esquema de controle de congestionamento do TCP responde bem ao aumento de tráfego em uma inter-rede. Desacelerando rapidamente, o TCP pode aliviar o congestionamento. Em sua essência, o TCP evita adicionar retransmissões a uma inter-rede congestionada. Mais importante, se todos os TCPs seguirem o padrão, o esquema de controle faz com que todos os remetentes se afastem quando ocorre um congestionamento, evitando o colapso.

25.13 Formato do segmento de TCP

O TCP usa um único formato para todas as mensagens, incluindo as que levam dados, as que levam acknowledgments e as que são parte do 3-way handshake usado para criar ou terminar uma conexão. O TCP emprega o termo *segmento* para se referir a uma mensagem; a Figura 25.6 mostra o formato do segmento.

Para entender o formato de segmento, é necessário lembrar que uma conexão de TCP contém duas *streams* de dados, uma fluindo em cada direção. Se os aplicativos em cada lado estão enviando dados simultaneamente, o TCP pode enviar um único segmento que transporta o acknowledgment para os dados que são recebidos, um anúncio de janela que especifica a quantidade de espaço de buffer adicional disponível para dados que chegam, e dados que estão sendo enviados. Deste modo, alguns campos no segmento se referem à stream de dados que viaja para frente, enquanto outros se referem à stream de dados que viaja na direção contrária.

Quando um computador envia um segmento, os campos *ACKNOWLEDGMENT NUMBER* e *WINDOW* se referem a dados que chegam: o *ACKNOWLEDGMENT NUMBER* especifica o número de seqüência dos dados que estão sendo esperados e *WINDOW* especifica quanto espaço de buffer adicional está disponível para mais dados começando na posição definida pelo *acknowledgment*². O campo *SEQUENCE NUMBER* se refere a dados sendo enviados, fornecendo o número de seqüência para o primeiro bit de dados que está sendo transportado no segmento. O receptor usa o número de seqüência para reordenar os segmentos que chegam fora de ordem e para computar um número de acknowledgment. O campo *DESTINATION PORT* identifica que programas aplicativos

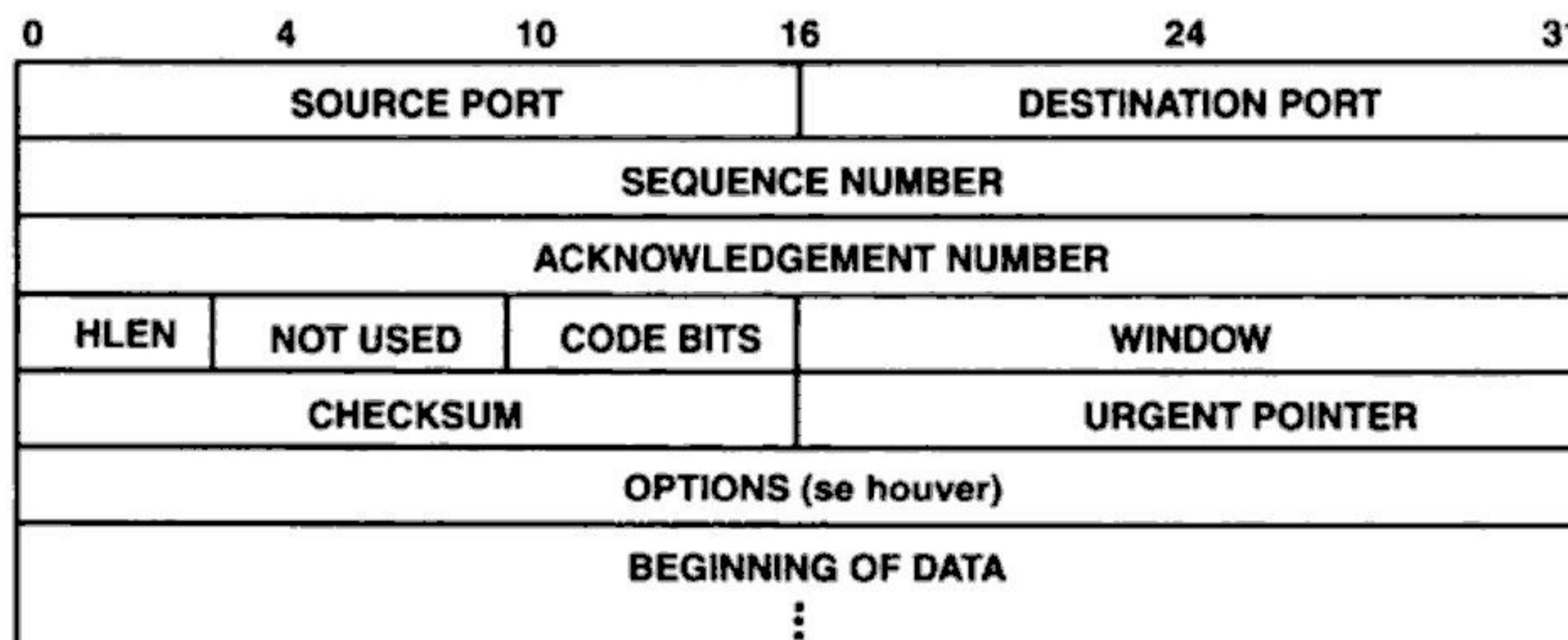


Figura 25.6 O formato do segmento do TCP. Cada mensagem enviada de uma máquina TCP para um TCP em outra (tanto de dados quanto de acknowledgments) usa este formato.

² O tamanho do acknowledgment e o tamanho da janela sempre fazem referência à primeira posição para a qual estão faltando dados; se segmentos chegam fora de ordem, um TCP receptor gera o mesmo acknowledgment múltiplas vezes até que cheguem os dados perdidos.

inicial? Quantas vezes o TCP tenta novamente antes de declarar que não pode abrir a conexão? Para descobrir, tente abrir uma conexão com um endereço inexistente e use um analisador de rede para acompanhar o tráfego de TCP resultante.

- 25.11** Escreva um programa de computador que compute o número máximo de conexões TCP abertas simultaneamente em um determinado momento (Experimento 12.3 em *Hands-On Networking*).



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

uma conexão com um ISP, do que agir como um único sistema autônomo com uma conexão para o resto da Internet. Mais importante, a seleção de um protocolo de roteamento pode determinar se uma organização escolhe usar múltiplos sistemas autônomos – o protocolo pode limitar o tamanho máximo da rede ou gerar tráfego de roteamento excessivo quando usado por um grande número de roteadores (ou seja, o tráfego de roteamento pode crescer como o quadrado do número de roteadores).

Para resumir:

Os roteadores na Internet global são divididos em quatro grupos, sendo cada um conhecido como um sistema autônomo. Os roteadores dentro de um sistema autônomo trocam informações de roteamento, que são, então, resumidas antes de serem repassadas a outro grupo.

27.7 Os dois tipos de protocolos de roteamento

Agora que é possível entender o conceito de sistemas autônomos, o roteamento da Internet pode ser definido mais precisamente. Todos os protocolos de roteamento da Internet se encaixam em uma de duas categorias. Depois de definir as duas categorias, serão examinados protocolos específicos em cada uma.

27.7.1 Protocolos de roteador interno (Internal Gateway Protocols, IGPs)

Os roteadores dentro de um sistema autônomo usam um *Protocolo de Roteador Interno (Internal Gateway Protocol, IGP)* para trocar informações de roteamento. Existem diversos IGPs disponíveis; cada sistema autônomo está livre para escolher seu próprio IGP. Usualmente, um IGP é fácil de instalar e de operar, mas pode limitar o tamanho ou a complexidade de roteamento de um sistema autônomo.

27.7.2 Protocolos de roteador externo (External Gateway Protocols, EGPs)

Um roteador em um sistema autônomo usa um *Protocolo de Roteador Externo (External Gateway Protocol, EGP)* para trocar informações de roteamento com um roteador em outro sistema autônomo. EGPs são geralmente mais complexos de instalar e operar do que IGPs, mas oferecem mais flexibilidade e menor sobre peso (ou seja, menos tráfego). Para economizar tráfego, um EGP resume a informação de roteamento de um sistema autônomo antes de passá-la para outro sistema autônomo. Mais importante, um EGP implementa *policy constraints* (restrições nas políticas permitidas) que permitem a um administrador de sistema determinar exatamente que informações são liberadas fora da organização.

27.7.3 Quando são usados IGPs e EGPs

A Figura 27.3 ilustra a hierarquia de roteamento de dois níveis usada na Internet global.

Na figura, o Sistema Autônomo 1 (AS_1) escolheu o IGP₁ para usar internamente, e o Sistema Autônomo 2 (AS_2) escolheu o IGP₂. Todos os roteadores no AS_1 se comunicam usando o IGP₁, e todos os roteadores em AS_2 se comunicam usando o IGP₂. Os roteadores R₁ e R₄ usam um EGP para se comunicar entre os dois sistemas autônomos. Isto é, o R₁ precisa resumir a informação do sistema autônomo e enviar o resumo para R₄. Além disto, o R₁ aceita um sumário de R₄ e usa o IGP₁ para propagar a informação para roteadores em AS_1 . O R₄ realiza o mesmo serviço para o AS_2 .

27.7.4 As melhores rotas, métricas de roteamento e IGPs

Pode parecer que, em vez de meramente descobrir um caminho para cada destino, softwares de roteamento devem encontrar todos os caminhos possíveis e então escolher o melhor. Embora a Inter-

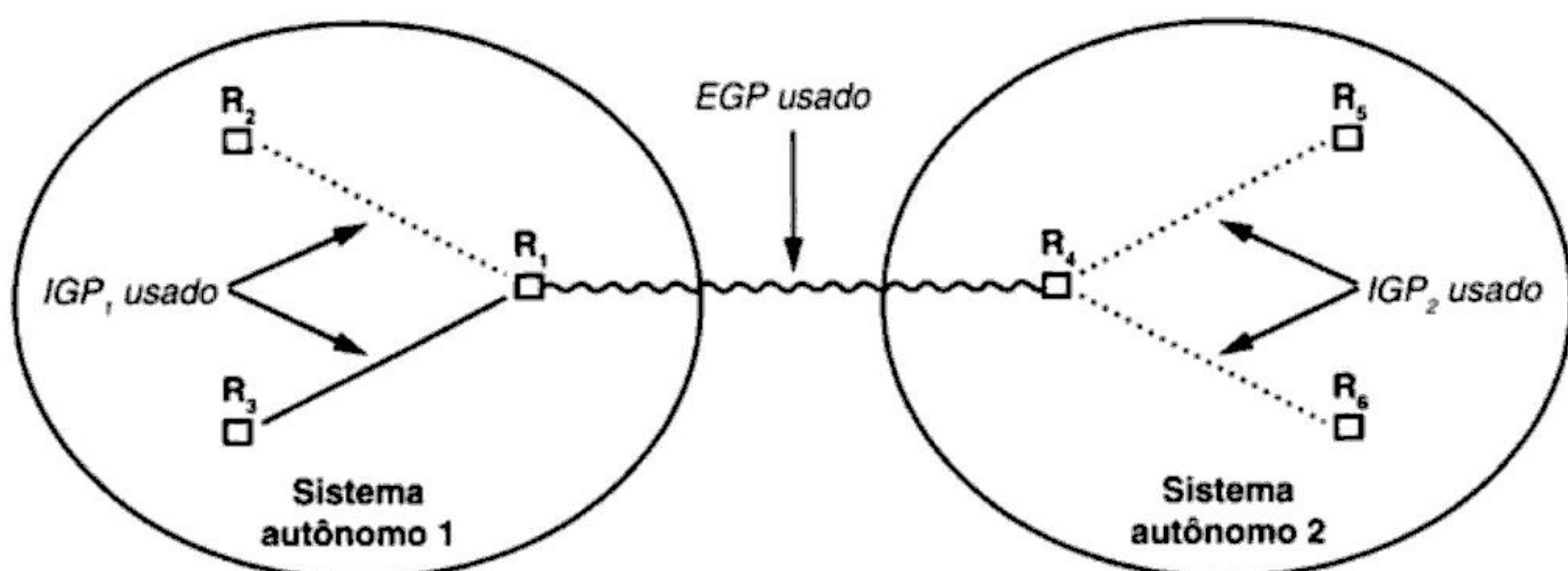


Figura 27.3 A arquitetura de roteamento da Internet. Cada sistema autônomo escolhe um IGP para usar internamente; um EGP é usado para a comunicação entre sistemas autônomos.

net global usualmente tenha múltiplos caminhos entre uma fonte e um destinatário, não há acordo universal sobre qual seria o melhor caminho. Para entender por quê, considere os requerimentos de vários aplicativos. Para um aplicativo de login interativo, o caminho com menor atraso é o melhor. Para um navegador fazendo download de um grande arquivo de gráficos, o caminho com maior desempenho é o melhor. Para um aplicativo de webcast de áudio que recebe áudio em tempo real, o caminho com menor jitter é o melhor.

O termo *métrica de roteamento* é usado para se referir à medição do caminho que o software de roteamento usa quando escolhe uma rota. Embora seja possível usar o desempenho, o atraso ou o jitter como métricas de roteamento, a maioria dos softwares de roteamento de Internet não o faz. Em vez disso, o roteamento de Internet típico usa uma combinação de duas métricas: *custo administrativo* e *conta de hops*. No roteamento da Internet, um hop corresponde a uma rede intermediária (ou um roteador). Por isso, a conta de hops para um destinatário dá o número de redes intermediárias no caminho para o destinatário. Os custos administrativos são atribuídos manualmente, seguidamente para controlar quais caminhos o tráfego pode utilizar. Por exemplo, suponha que em uma corporação dois caminhos conectem o departamento de contas ao departamento de pagamentos: um caminho de 2 hops que inclui uma rede destinada para o tráfego de clientes e um caminho de 3 hops que inclui redes para o tráfego interno da corporação. Isto é, o caminho mais curto viola a política da corporação por atravessar uma rede destinada para servir os clientes. Em tais casos, uma rede administrativa pode sobrepor-se ao custo real do caminho de 2 hops atribuindo ao caminho um custo administrativo de 4 hops (ou seja, o administrador substitui o custo real por um valor administrativo para alcançar o efeito desejado). O software de roteamento irá escolher o caminho com o menor custo (ou seja, o caminho com uma métrica de 3 hops). Por isso, o tráfego irá seguir a política da corporação.

Os IGPs e os EGPs diferem de maneira importante no que diz respeito às métricas de roteamento: IGPs usam métricas de roteamento, mas os EGPs não. Isto é, cada sistema autônomo escolhe uma métrica de roteamento e organiza o software de roteamento interno para que ele envie as métricas com cada rota, de forma que os softwares receptores possam usar a métrica para escolher o melhor caminho. Fora de um sistema autônomo, contudo, um EGP não tenta encontrar o melhor caminho. Em vez disso, o EGP meramente escolhe um caminho. A razão é simples: como cada sistema autônomo está livre para escolher uma métrica de roteamento, um EGP não consegue fazer comparações significativas. Por exemplo, suponha que um sistema autônomo relate o número de hops através de um caminho para o destinatário D e outros sistemas autônomos relatam o throughput em um caminho diferente para D . Um EGP que recebe os dois relatórios não pode escolher qual dos dois caminhos têm menor custo porque não há como converter de hops

- *Estruturas para roteamento de trânsito.* O BGP classifica cada sistema autônomo como sistema de tráfego se ele concorda em deixar passar tráfego através dele para outro sistema autônomo, ou como sistema *stub*, se ele não deixa. Similarmente, o tráfego passando através para outro AS é classificado como tráfego de trânsito. A classificação permite ao BGP distinguir entre ISPs e outros sistemas autônomos. Mais importante, o BGP permite a uma corporação classificar a si própria como um *stub* mesmo que ela seja *multi-homed* (ou seja, uma corporação com múltiplas conexões externas pode se recusar a aceitar tráfego de trânsito).
- *Transporte confiável.* O BGP usa o TCP para todas as comunicações. Isto é, um programa BGP em um roteador num sistema autônomo forma uma conexão TCP até um programa BGP em um outro sistema autônomo, e então envia dados através dessa conexão. O TCP garante que os dados chegam na ordem correta e que nenhum deles falta.

O BGP é especialmente importante na Internet global porque alguns dos maiores ISPs usam o BGP para trocar informações de roteamento. Em particular, ISPs que se interconectam em pontos de acesso de redes usam BGPs. Para garantir que o roteamento permaneça consistente, diversas organizações tentam manter uma base de dados de todas as rotas. Por exemplo, uma organização chamada RIPE (*Resaux IP Europeens*) mantém um *registro de roteamento* que contém uma lista de destinatários na Internet e informações sobre o ISP de cada destinatário. Para resumir:

A versão 4 do Protocolo de Portal Fronteiriço (BGP-4) é o Protocolo de Portal Externo usado para trocar informações de roteamento entre sistemas autônomos na Internet global. ISPs usam o BGP-4 para obter informações de roteamento entre si. Para garantir que um datagrama de um computador arbitrário para um destinatário arbitrário seja encaminhado corretamente, a informação de roteamento global precisa ser consistente.

27.10 O Routing Information Protocol (RIP)

Embora o BGP-4 sustente a troca de roteamentos entre sistemas autônomos, suporte adicional de Protocolos de Portal Interno é necessário para permitir aos roteadores passar essa informação dentro de um sistema autônomo. Serão examinados dois desses protocolos. Esta seção examina o *Protocolo de Roteamento de Informação (Routing Information Protocol, RIP)*; a próxima seção examina o *OSPF*.

O RIP estava entre os primeiros protocolos de roteamento usados com o IP. O protocolo é implementado pelo programa *roteado (routed)*¹ que vem com a maioria dos sistemas UNIX. O RIP tem as seguintes características.

- *Roteamento dentro de um sistema autônomo.* O RIP foi projetado como um Protocolo de Roteamento Interno usado para passar informações entre roteadores dentro de um sistema autônomo.
- *Métrica de conta de hops.* O RIP mede a distância em *hops* de rede, em que cada rede entre a fonte e o destino conta como um único hop. Além disso, o RIP usa a contagem de origem em 1, o que significa que uma rede conectada diretamente está a um hop de distância, e não zero.
- *Transporte não-confiável.* O RIP usa o UDP pra toda transmissão de mensagem.
- *Entrega de broadcast e multicast.* O RIP é projetado para o uso em tecnologias de Redes Locais que suportam broadcast ou multicast (por exemplo, a Ethernet). A versão 1 do RIP usa broadcast de hardware quando envia mensagens entre roteadores; a versão 2 permite a entrega através de multicast.

¹ O nome do programa, em inglês, é pronunciado "route dee".



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

Um grupo de multicast IP é anônimo de duas maneiras. Primeiro, nem um emissor nem um receptor sabem (ou podem descobrir) a identidade ou o número de membros no grupo. Segundo, roteadores e hosts não sabem quais aplicativos irão enviar um datagrama para um grupo, pois um aplicativo arbitrário pode enviar um datagrama a qualquer grupo de multicast a qualquer hora. Isto é, a associação em um grupo multicast define apenas um conjunto de receptores – um emissor não precisa entrar em um grupo de multicast para enviar uma mensagem a ele.

Para resumir:

A associação em um grupo de multicast IP é dinâmica: um computador pode entrar ou deixar o grupo a qualquer hora. A associação ao grupo define um conjunto de receptores; um aplicativo arbitrário pode enviar um datagrama ao grupo mesmo sem ser um membro do grupo.

27.15.2 IGMP

Como um host entra ou sai de um grupo de multicast? Existe um protocolo padrão que permite a um host informar um roteador próximo quando ele precisa entrar ou sair de um grupo de multicast em particular. Conhecido como *Protocolo Multicast de Grupos da Internet (Internet Group Multicast Protocol, IGMP)*, o protocolo é usado unicamente na rede entre um host e um roteador. Além disso, o protocolo define o computador, não o aplicativo, para ser membro do grupo. Se múltiplos aplicativos em um determinado computador entram em um grupo de multicast, o computador precisa fazer cópias de cada datagrama que recebe para o grupo. Apenas quando o último aplicativo sai de um grupo o computador usa o IGMP para informar o roteador local de que ele não é mais um membro do grupo.

27.15.3 Técnicas de encaminhamento e descoberta

Quando aprende que um host em uma de suas rede entrou em um grupo de multicast, o roteador precisa estabelecer um caminho para esse grupo e propagar datagramas que ele recebe para o grupo ao host. Por isso roteadores, não hosts, tem a responsabilidade pela propagação da informação de roteamento multicast.

A associação e o sustento para emissores anônimos em grupos dinâmicos torna o roteamento de multicast para propósitos genéricos extremamente difícil. Além disso, o tamanho e a topologia dos grupos variam consideravelmente entre aplicativos. Por exemplo, uma teleconferência seguidamente cria pequenos grupos (entre 2 e 5 membros) que podem estar geograficamente dispersos ou na mesma organização. Um aplicativo como o do tipo webcasting pode criar um grande grupo que se expande pelo globo.

Para acomodar as associações dinâmicas, os protocolos de roteamento multicast precisam ser capazes de mudar o roteamento rápida e continuamente. Por exemplo, se um usuário na França entra em um grupo de multicast que tem membros nos Estados Unidos e no Japão, o software de roteamento multicast precisa primeiro encontrar outros membros do grupo, e então criar a melhor estrutura de encaminhamento possível. Mais importante, como um usuário arbitrário pode enviar um datagrama ao grupo, o roteamento deve ser estendido além dos membros do grupo. Na prática, os protocolos multicast seguem três aproximações diferentes para o encaminhamento de datagramas:

- *Flood-and-prune (enchente-e-podagem)*. É ideal em uma situação em que o grupo é pequeno e todos os membros estão ligados a Redes Locais contíguas (por exemplo, um grupo dentro de uma corporação). Inicialmente, os roteadores encaminham cada datagrama para todas as redes. Isto é, quando chega um datagrama multicast, o roteador transmite-o em todas as LANs diretamente ligadas a ele via multicast de hardware. Para evitar loops de roteamento, os protocolos de enchente-e-podagem usam uma técnica conhecida como *Broadcast de Caminho Reverso (Reverse Path Broadcast, RPB)*, que quebra os ciclos. Enquanto o estágio de



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

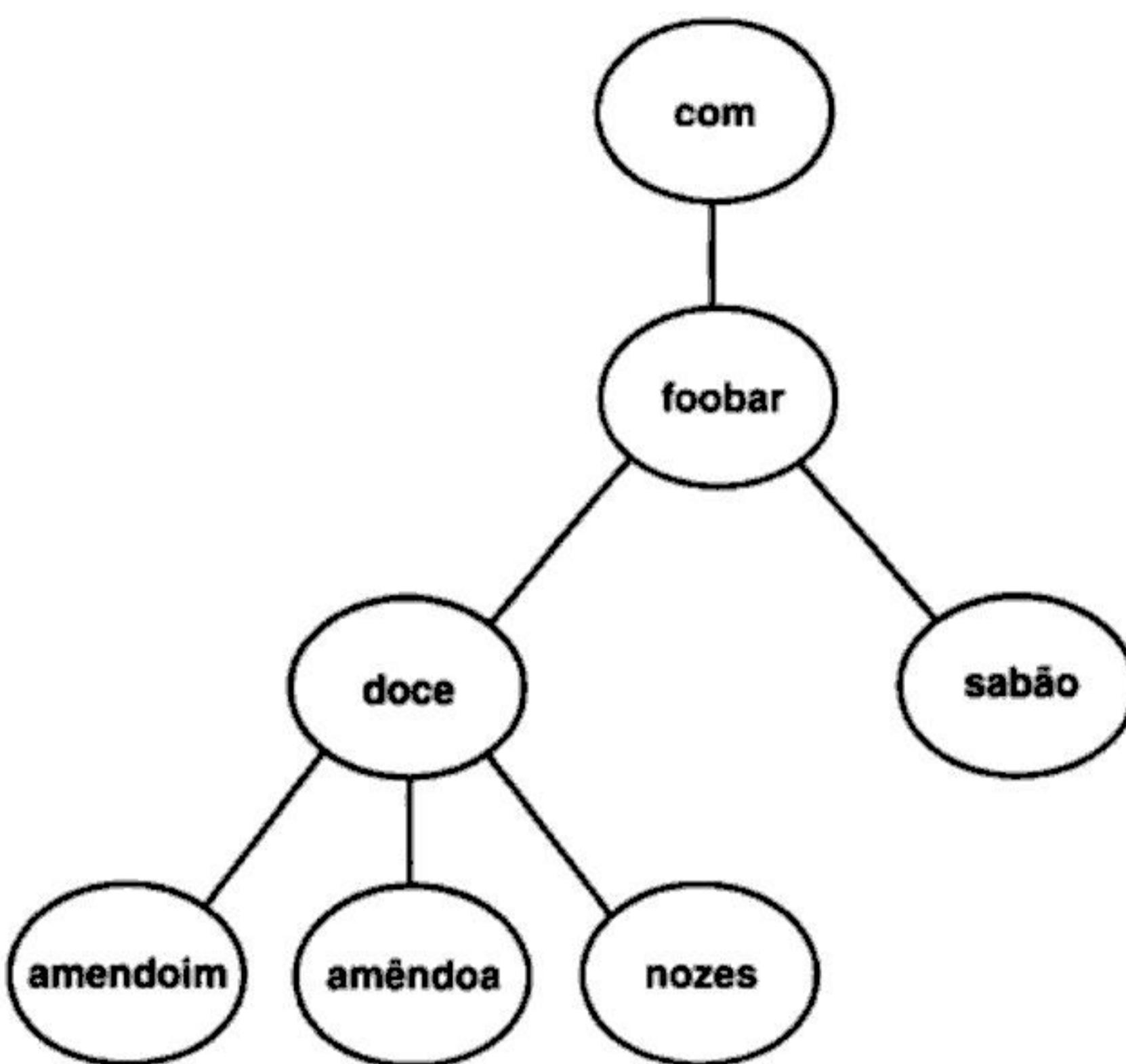


Figura 31.2 Uma representação visual que ilustra uma maneira que a hierarquia de DNS poderia ser estruturada em uma corporação. Os nomes de computadores individuais também poderiam ser acrescentados ao diagrama.

31.5 Os nomes de domínio que começam com www

Embora um nome do domínio denote um computador, muitas organizações atribuem nomes do domínio que refletem o serviço fornecido pelo computador. Por exemplo, o capítulo 34 discute o *File Transfer Protocol (FTP)*. Se a corporação Foobar fornece serviço FTP, ela pode escolher um computador para rodar este serviço e atribuir-lhe o seguinte nome de domínio:

ftp.foobar.com

Similarmente, a um computador que roda um servidor Web pode ser atribuído o nome:

www.foobar.com

Embora nomes descritivos sejam fáceis para os humanos de lembrar e usar, não são obrigatórios. O uso do www para nomear os computadores que rodam o servidor Web é meramente uma convenção. Um computador arbitrário pode rodar um servidor web, mas o nome do domínio deste computador não necessita conter www. Além disto, um computador que tem um nome do domínio começando por www não obriga o mesmo rodar um servidor Web. O ponto é:

Usar o primeiro rótulo no nome do domínio para denotar o serviço que o computador oferece é meramente uma convenção para ajudar os humanos – um computador que roda um servidor Web não necessita ser chamado www, assim como um computador chamado www não necessita rodar um servidor Web.

31.6 O modelo cliente-servidor do DNS

Uma das características principais do Domain Naming System é a autonomia – o sistema foi projetado para permitir que cada organização atribua nomes a seus computadores ou que mude esses

Organizações maiores usualmente descobrem que um único servidor centralizado não basta por duas razões. Primeiro, um único servidor e o computador em que ele executa não pode tratar requisições arbitrárias em alta velocidade. Segundo, grandes organizações freqüentemente acham difícil administrar uma base de dados centralizada. O problema é especialmente severo porque a maioria dos softwares de DNS não fornece atualização automatizada – uma pessoa deve entrar manualmente as mudanças e adições na base de dados do servidor². Deste modo, o grupo de pessoas responsável pela administração do servidor centralizado de uma organização deve se coordenar para assegurar que somente um gerente tente aplicar mudanças em um determinado momento. Se a organização executa múltiplos servidores, cada grupo pode administrar um servidor que é uma autoridade para os computadores do grupo. Mais importante, cada grupo pode fazer mudanças na base de dados de seu servidor sem a coordenação central.

31.9 Localidade de referência e múltiplos servidores

O princípio de referência de localidade discutido no Capítulo 7 se aplica ao sistema de nomes de domínios e ajuda a explicar por que múltiplos servidores funcionam bem. O sistema de nomes de domínios segue o princípio de referência de localidade em dois modos. Primeiro, um usuário tende a pesquisar os nomes de computadores locais mais freqüentemente que os nomes de computadores remotos. Segundo, um usuário tende a pesquisar repetidamente o mesmo conjunto de nomes de domínio. Ter múltiplos servidores dentro de uma organização funciona bem porque um servidor pode ser colocado dentro de cada grupo. O servidor local é uma autoridade para nomes de computadores no grupo. Porque o DNS obedece ao princípio de localidade, o servidor local pode tratar a maioria das requisições. Deste modo, além de ser mais fácil de administrar, múltiplos servidores ajudam a equilibrar a carga, reduzindo, assim, os problemas de disputa (*contention*) que um servidor centralizado pode causar.

31.10 Links entre servidores

Embora o DNS dê a liberdade de usar múltiplos servidores, uma hierarquia de domínio não pode ser dividida em servidores arbitrariamente. A regra é: um único servidor deve ser responsável por todos os computadores que têm um determinado sufixo. Em termos da representação visual, subárvores podem ser movidas para um servidor separado, mas um determinado nó não pode ser dividido.

Os servidores no sistema de nomes de domínios são ligados juntos, possibilitando que um cliente encontre o servidor correto seguindo os links. Em particular, cada servidor é configurado para conhecer as localizações de servidores de subpartes da hierarquia. Por exemplo, nos dois arranjos de servidor mostradas na Figura 31.3, o servidor para *.com* deve ser configurado para conhecer a localização do servidor para *foobar.com*. Além disso, o servidor para *foobar.com* deve ser configurado para conhecer a localização de outros servidores. Por exemplo, na Figura 31.3b, o servidor para *foobar.com* seria configurado para conhecer a localização do servidor para *nozes.doces.foobar.com*. Finalmente, cada servidor DNS é configurado para conhecer a localização de um servidor raiz. Para resumir:

Todos servidores de nome de domínio são ligados juntos para formar um sistema unificado. Cada servidor sabe como alcançar um servidor raiz e como alcançar servidores que são autoridades para nomes mais abaixo na hierarquia.

² O IETF está trabalhando em um padrão para atualização automatizada de uma base de dados de um servidor DNS.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

Transferência de Arquivos e Acesso a Arquivos Remotos

34.1 Introdução

Os capítulos anteriores desta seção definem o paradigma cliente-servidor e dão um exemplo de aplicativo de rede. Este capítulo apresenta outro exemplo: um aplicativo de rede que pode transferir cópias de um arquivo de um computador para outro. Além de discutir a interface de transferência de arquivos, o capítulo considera o acesso a arquivos e explica como o software subjacente usa o paradigma cliente-servidor.

34.2 Transferência de arquivos generalizada

Um *arquivo (file)* é a abstração fundamental para o armazenamento por longos períodos. Conforme as redes surgiam, logo se tornou aparente que um único protocolo generalizado podia ser projetado para trabalhar com muitos aplicativos. O problema se tornou conhecido como problema da *transferência de arquivos*, e um sistema de software que movia dados arbitrários de um arquivo em um computador para um arquivo em outro se tornou conhecido como software de *transferência de arquivos*.

A transferência de arquivos é complicada porque uma inter-rede pode conectar sistemas de computador heterogêneos. Portanto, o software de transferência de arquivos deve acomodar diferenças entre os sistemas de computador na maneira que armazenam arquivos. Por exemplo, cada sistema de computador tem regras sobre nomes de arquivos – um nome válido em um sistema de computador pode ser inválido em outro. Além disso, extensões de arquivo que denotam o tipo podem diferir (por exemplo, um computador usa a extensão .jpeg para imagens JPEG e outro usa a extensão .jpg). Uma vez que a maioria dos sistemas de computador usa contas de login para definir propriedades de arquivo, o proprietário em um sistema de computador pode não ter uma conta de login correspondente em outro. Finalmente, o software de transferência de arquivos deve acomodar outras diferenças menores nas representações de arquivo, informações de tipo e mecanismos de proteção de arquivos. Para resumir:

Um serviço de transferência de arquivos pode mover uma cópia de um arquivo de um computador para outro. A transferência de arquivos é complicada pelas diferenças na forma como os computadores nomeiam e armazenam arquivos.

34.3 O protocolo de transferência de arquivos

O serviço de transferência de arquivos mais extensamente empregado na Internet usa o *File Transfer Protocol (FTP)*. As características do FTP são:

- *Propósito Geral.* O FTP trata de muitos dos conceitos discutidos acima.
- *Conteúdo de Arquivo Arbitrário.* O FTP permite a transferência de dados arbitrários.
- *Autenticação e Propriedade.* O FTP inclui um mecanismo que permite que os arquivos tenham propriedade e restrições de acesso.
- *Acomoda a Heterogeneidade.* O FTP esconde os detalhes dos sistemas de computador individuais – ele pode ser usado para transferir uma cópia de um arquivo entre um par arbitrário de computadores.

O FTP está entre os protocolos de aplicativo mais antigos ainda em uso na Internet e é invocado pelos navegadores quando um usuário requer um *download* de arquivo. Originalmente definido como parte dos protocolos da ARPANET, o FTP precede TCP e IP. Com a criação do TCP/IP, uma nova versão do FTP foi desenvolvida para funcionar com os novos protocolos da Internet.

O FTP é muito usado – no início da história da Internet, datagramas carregando transferências de arquivos eram responsáveis por aproximadamente um terço de todo tráfego da Internet; o tráfego gerado por serviços como e-mail e o sistema de nomes de domínios não chegaram perto de exceder aquele gerado pelo FTP¹.

Para resumir:

O serviço de transferência de arquivos mais popular na Internet usa FTP, o File Transfer Protocol. O FTP é um protocolo de propósito geral que pode ser usado para copiar um arquivo arbitrário de um computador para outro.

34.4 Modelo de FTP e interface com o usuário geral

O FTP é projetado para ser executado a partir de um programa (por exemplo, um navegador) ou para uso interativo. Quando invoca o FTP, o programa precisa tratar de todos os detalhes e então informar ao usuário se a operação teve sucesso ou falhou; o usuário nunca vê a interface de FTP. Quando um usuário invoca o FTP interativamente, ele se comunica com uma interface dirigida por comandos. O FTP emite um *prompt* ao qual o usuário responde entrando um comando. O FTP executa o comando e então emite outro prompt.

O FTP tem comandos que permitem a um usuário especificar um computador remoto, fornecer autorização, descobrir que arquivos remotos estão disponíveis e requisitar a transferência de um ou mais arquivos. Alguns comandos de FTP exigem pouco ou nenhum tempo para executar, enquanto outros podem levar um tempo significativo. Por exemplo, pode levar muitos segundos para transferir uma cópia de um arquivo grande.

34.5 Comandos do FTP

Embora o protocolo padrão de FTP especifique exatamente como o software de FTP em um computador interage com o software de FTP em outro, ele não especifica uma interface com o usuário.

¹ Em 1995, o tráfego com informações da World Wide Web ultrapassou o FTP pela primeira vez.

Conseqüentemente, a interface disponível para um usuário pode variar de uma implementação de FTP para outra. Para ajudar a manter a semelhança entre produtos, muitos vendedores escolheram adotar a interface que primeiro apareceu em uma versão inicial do software de FTP escrito para o sistema BSD UNIX.

A interface BSD para o FTP suporta mais de 50 comandos individuais. A Figura 34.1 lista os nomes dos comandos.

!	cr	macdef	proxy	sendport
\$	delete	mdelete	put	status
account	debug	mdir	pwd	struct
append	dir	mget	quit	sunique
ascii	disconnect	mkdir	quote	tenex
bell	form	mls	recv	trace
binary	get	mode	remotehelp	type
bye	glob	mput	renam	user
case	hash	nmap	reset	verbose
cd	help	ntrans	rmdir	?
cdup	lcd	open	runique	
close	ls	prompt	send	

Figura 34.1 Os nomes dos comandos encontrados na interface BSD para FTP. Muitos vendedores suportam uma variante da interface do BSD.

A lista de comandos pode parecer opressiva para um novato por duas razões. Primeiro, a interface do BSD contém opções que raramente são implementadas (por exemplo, o comando *proxy*, que permite a comunicação simultânea com dois sites remotos). Segundo, a interface fornece várias opções para tratar de detalhes obscuros, alguns dos quais já se tornaram irrelevantes. Por exemplo, *tenex* se refere a um sistema operacional obsoleto.

O FTP acomoda também múltiplas representações de arquivos de texto. Por exemplo, alguns sistemas de computador usam um único caractere de *avanço de linha (linefeed)* para separar linhas de texto. Outros sistemas usam uma seqüência de dois caracteres que consiste em um *retorno de carro (carriage return)* seguido de um *avanço de linha*. O comando *cr* permite a um usuário especificar que representação a máquina remota usa, tornando possível para o FTP traduzir entre as representações remotas e locais.

Uma forma final de complexidade advém porque a interface de BSD inclui também nomes alternativos (isto é, múltiplos nomes para uma função). Por exemplo, qualquer um dos comandos *close* ou *disconnect* pode ser usado para terminar uma conexão com um computador remoto. De forma semelhante, *bye* ou *quit* podem ser usados para terminar o programa de FTP, e *help* ou *?* podem ser usados para obter uma lista de comandos disponíveis.

34.6 Conexões, autorização e permissões de arquivo

Felizmente, apenas alguns comandos de FTP são necessários para transferir um arquivo. Depois de começar um programa de FTP, um usuário deve digitar o comando *open* antes que quaisquer arquivos possam ser transferidos. *Open* exige que o usuário forneça o nome de domínio de um computador remoto e então forma uma conexão de TCP com o computador.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

A etiqueta correspondente usada para terminar uma operação tem o seguinte formato:

```
| </TAGNAME>
```

Por exemplo, um documento HTML começa com a etiqueta `<HTML>`. O par de etiquetas `<HEAD>` e `</HEAD>` cercam o cabeçalho, enquanto o par de etiquetas `<BODY>` e `</BODY>` cercam o corpo. No cabeçalho, as etiquetas `<TITLE>` e `</TITLE>` cercam o texto que forma o título do documento. A Figura 35.1 mostra a forma geral de um documento HTML.

```
<HTML>
  <HEAD>
    <TITLE>
      o texto que forma o título do documento
    </TITLE>
  </HEAD>
  <BODY>
    o corpo do documento aparece aqui
  </BODY>
</HTML>
```

Figura 35.1 A forma geral de um documento HTML. O cabeçalho contém informações sobre o documento; o corpo contém o documento propriamente dito.

Na figura, cada etiqueta aparece em uma nova linha, e indentação é usada para mostrar a estrutura. Porém, tais convenções são importantes somente para as pessoas lerem o documento – um navegador ignora todo esse espaçamento. Deste modo, o documento HTML na Figura 35.2 é equivalente ao documento na Figura 35.1.

```
<HTML><HEAD><TITLE>texto que forma o
título do documento
</TITLE></HEAD><BODY>corpo do documento aparece
aqui</BODY></HTML>
```

Figura 35.2 O documento HTML da Figura 35.1 com alguns dos espaços em branco desnecessários removidos. Um navegador produz a mesma saída para ambos os documentos.

35.6 Exemplos de etiquetas HTML de formatação

Porque é armazenado em um arquivo texto, o documento HTML deve conter etiquetas explícitas que especifiquem como a saída deve ser exibida. Por exemplo, a etiqueta `
` instrui um navegador para introduzir uma quebra de linha. Isto é, quando encontra um `
` na entrada, o navegador move para o início da próxima linha na tela antes de gerar mais saída. Deste modo, a seguinte seqüência de HTML:

```
Olá.<BR>Este é um exemplo<BR>de HTML.
```

faz com que um navegador apresente três linhas de saída:

Olá.

Este é um exemplo
de HTML.

Duas quebras em seqüência deixam uma linha em branco na saída. Deste modo,

Olá.

Isto mostra o
 espaçamento no HTML.

resulta na seguinte saída:

Olá.

Isto mostra o
espaçamento no HTML.

35.7 Cabeçalhos

O HTML inclui seis pares de etiquetas que podem ser usadas para exibir cabeçalhos na saída. Uma etiqueta da forma `<Hi>` marca o começo de um cabeçalho de nível *i*, e uma etiqueta da forma `</Hi>` marca o final. Por exemplo, o texto para o nível mais importante de cabeçalho é cercado por `<H1>` e `</H1>`. Browsers normalmente exibem o texto do cabeçalho de nível 1 no maior tamanho, cabeçalhos de nível 2 ligeiramente menores, e assim por diante. Deste modo, quando um navegador mostra a entrada:

Olá.
<H1>Este é um cabeçalho</H1>
De volta ao normal.

o navegador escolherá um tamanho de ponto grande para o cabeçalho:

Olá.

Este é um Cabeçalho

De volta ao normal.

35.8 Listas

Além de cabeçalhos, o HTML permite que um documento contenha listas. A forma mais simples é uma *lista não ordenada*, a qual solicita ao navegador que mostre uma lista de itens³. Na especificação HTML, as etiquetas `` e `` cercam a lista inteira, e cada item da lista começa com a etiqueta ``. Normalmente, um navegador coloca um *bullet* na frente de cada item. Por exemplo, a entrada:

Aqui está uma lista de 5 nomes:

 Scott Sharon Jan

³ O termo *não ordenado* significa que o navegador não necessita classificar a lista de acordo com um número de seqüência.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

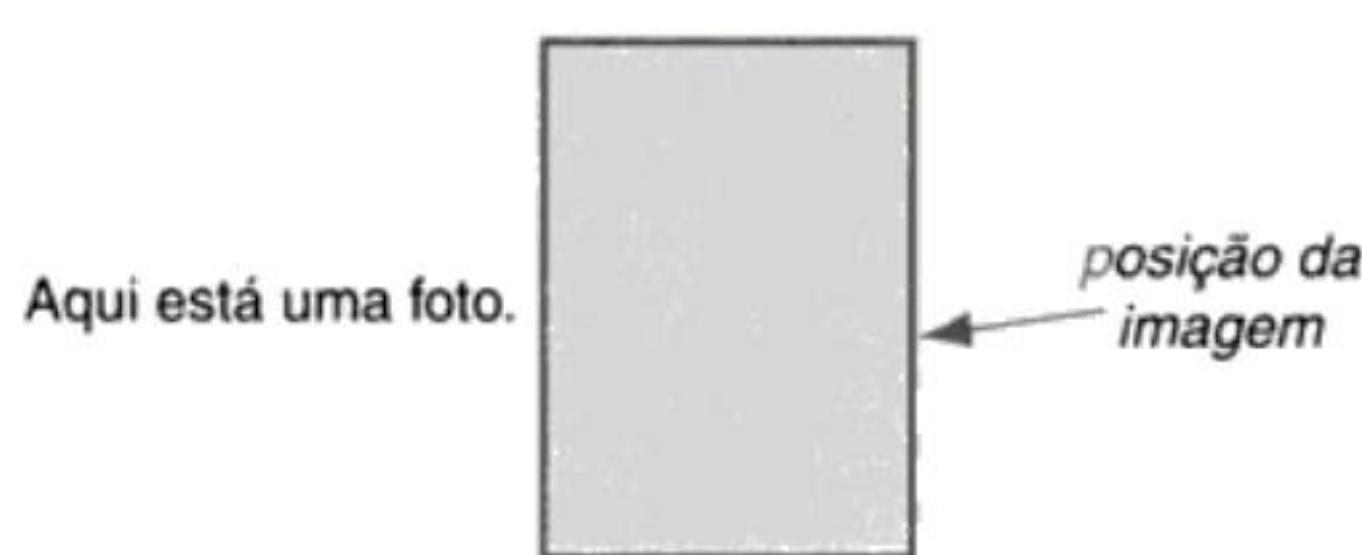


Figura 35.3 Ilustração de alinhamento de imagem. Como solicitado na etiqueta, o texto na linha é posicionado no meio da imagem.

ter muitas páginas; cada uma deve receber um nome único. Terceiro, a Web suporta múltiplas representações de documento, então um navegador deve saber qual representação uma página usa (por exemplo, se o nome refere-se a um documento HTML ou a uma imagem armazenada em formato binário). Quarto, porque a Web é integrada com outros aplicativos, um navegador deve saber qual protocolo de aplicativo deve ser usado para acessar uma página.

Foi inventado um formato sintático que incorpora todas as informações necessárias para especificar um item remoto. A forma sintática codifica as informações em uma cadeia alfanumérica conhecida como *Uniform Resource Locator (URL)*. A forma geral de uma URL é:

| protocol://computer_name:port/document_name

onde *protocolo* é o nome do protocolo usado para acessar o documento, *computer_name* é o nome de domínio do computador em que o documento reside, *:port* é um número de porta de protocolo opcional⁴ e *document_name* é o nome do documento no computador especificado. Por exemplo, a URL

[http:// www.netbook.cs.purdue.edu/toc/toc01.htm](http://www.netbook.cs.purdue.edu/toc/toc01.htm)

especifica o protocolo *http*, um computador de nome *www.netbook.cs.purdue.edu* e um arquivo de nome *toc/toc01.htm*.

Como o exemplo mostra, uma URL contém as informações de que um navegador precisa para carregar uma página. O navegador usa os caracteres separadores dois pontos e barra para dividir a URL em três componentes: protocolo, nome de computador e nome de documento. O navegador então usa as informações para acessar o documento especificado.

35.11 Links de hipertexto de um documento para outro

Embora HTML inclua muitas características usadas para descrever o conteúdo e o formato de um documento, a característica que distingue HTML de linguagens de formatação de documentos convencionais é a habilidade de incluir referências do tipo hipertexto. Cada referência do tipo hipertexto é um ponteiro passivo para outro documento. Diferentemente de uma referência *IMG*, que faz com que um navegador carregue informações externas imediatamente, uma referência do tipo hipertexto não causa uma ação imediata. Em vez disso, um navegador transforma uma referência do tipo hipertexto em um item selecionável ao exibir o documento. Se o usuário seleciona o item, o navegador segue a referência, carrega o documento ao qual ela se refere e substitui a exibição corrente com o novo documento.

⁴ O número de porta de protocolo opcional raramente é usado; por isso, o omitiremos da discussão.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

grama CGI pode gerar texto puro ou uma imagem digital. Para distinguir entre vários tipos de documento, o padrão permite que um programa CGI coloque um cabeçalho em sua saída. O cabeçalho consiste em um texto que descreve o tipo de documento.

Os cabeçalhos gerados por um programa CGI usam o mesmo formato geral de cabeçalhos HTTP: o cabeçalho consiste em uma ou mais linhas de texto e uma linha em branco. Cada linha do cabeçalho especifica informações sobre o documento e a representação dos dados.

Depois de executar um programa CGI, um servidor examina o cabeçalho antes de retornar o documento ao navegador que emitiu a requisição. Deste modo, um programa CGI pode usar o cabeçalho para se comunicar com o servidor quando for necessário. Por exemplo, um cabeçalho que consiste na linha:

```
Content-type: text/html
```

seguida por uma linha em branco, especifica que a saída é um documento HTML.

O cabeçalho na saída de um programa CGI pode ser usado também para especificar que o documento está em uma nova localização. A técnica é conhecida como *redirecionamento*. Por exemplo, suponha que um programa CGI associado à URL:

```
http://someserver/cgi-bin/foo
```

precisa redirecionar requisições recebidas para o documento associado à URL:

```
http://someserver/new/bar.txt
```

O programa CGI pode gerar duas linhas de saída:

```
Location: /new/bar.txt
<linha em branco>
```

Quando detectar a diretiva *Location:* no cabeçalho, o servidor carregará o documento como se o navegador tivesse solicitado */new/bar.txt* (em vez de *cgi-bin/foo*) com a URL:

```
http://someserver/new/bar.txt
```

Podemos resumir:

A saída de um programa CGI pode começar com um cabeçalho que o servidor interpreta. Um cabeçalho pode especificar o formato do documento, ou que o documento solicitado foi movido para uma URL diferente.

36.7 Um programa CGI exemplo

A Figura 36.1 contém o código fonte para um programa CGI trivial escrito na linguagem de Shell UNIX. Quando executado, o programa cria um documento de texto que contém a data e a hora atuais.

Os scripts do Shell podem ser difíceis de entender. Essencialmente, o Shell é um interpretador de comandos – um script do Shell contém comandos no mesmo formato que um usuário usa ao entrar comandos via teclado. Com exceção da primeira linha, o Shell ignora linhas em branco ou qualquer linha que comece com o símbolo *#*. No exemplo, linhas que contêm um símbolo *#* são usadas para comentários, enquanto outras linhas contêm um comando válido. A primeira “palavra” de uma linha é o nome do comando; as palavras subsequentes formam argumentos para o comando.

```
#!/bin/sh

#
# script CGI que imprime a data e a hora em que foi executado
#
# Gera o cabeçalho do documento seguido por uma linha em branco
echo Content-type: text/plain
echo

# Saída o echo da data
echo Este documento foi criado em `date`
```

Figura 36.1 Um programa CGI exemplo escrito em linguagem Shell UNIX.

O único comando usado no script exemplo é *echo*. Cada invocação de *echo* gera uma linha de saída. Quando executado, o script invoca *echo* três vezes. Deste modo, o script gerará exatamente três linhas de saída.

Quando invocado sem argumentos, *echo* cria uma linha em branco. Caso contrário, escreve uma cópia exata de seus argumentos. Por exemplo, o comando:

```
echo smord hplar
```

gera uma linha de saída que contém duas palavras:

```
smord hplar
```

O único item incomum no script exemplo é a construção '*date*'. O Shell interpreta a aspa simples como uma requisição para executar o programa de data e substitui sua saída. Deste modo, antes de invocar *echo* pela terceira vez, o shell substitui a string '*date*' com a data e a hora atual. Como resultado, a terceira invocação de *echo* gera uma linha de saída que contém a data real. Por exemplo, se fosse executado no dia 3 de junho de 2004, 37 segundos após 2:19 da tarde, o script geraria a seguinte saída:

```
Content-type: text/plain
Este documento foi criado em Thu Jun 3 14:19:37 EST 2004
```

Um servidor capaz de executar programas CGI deve ser configurado antes de poder invocar o script exemplo. A configuração especifica uma URL que o servidor usa para localizar o script. Quando um navegador contata o servidor e solicita a URL especificada, o servidor executa o programa. Quando recebe o documento, o navegador remove o cabeçalho e mostra uma única linha para o usuário:

```
Este documento foi criado em Thu Jun 3 14:19:37 EST 2004
```

Diferentemente de um documento estático, o conteúdo do documento dinâmico muda cada vez que o usuário solicita ao navegador que recarregue o documento. Porque o navegador não armazena na cache um documento dinâmico, uma requisição para recarregar o documento faz com que o navegador contacte o servidor. O servidor invoca o programa CGI, que cria um novo documento com a hora e a data atuais. Deste modo, o usuário vê a mudança no conteúdo do documento após cada requisição de recarga.

36.8 Parâmetros e variáveis de ambiente

O padrão permite que um programa CGI seja parametrizado. Isto é, um servidor pode passar argumentos para um programa CGI sempre que o programa for invocado. A parameterização é importante porque permite que um único programa CGI trate de um conjunto de documentos dinâmicos que diferem apenas em detalhes secundários. Mais importante, valores para parâmetros podem ser fornecidos pelo navegador. Para fazê-lo, o navegador acrescenta informações adicionais a uma URL. Quando uma requisição chega, o servidor divide a URL na requisição em duas partes: um prefixo que especifica um documento particular e um sufixo que contém informações adicionais. Se o prefixo da URL corresponde a um programa CGI, o servidor invoca o programa e passa o sufixo da URL como um argumento.

Sintaticamente, um ponto de interrogação (?) na URL separa o prefixo do sufixo. Tudo que vem antes de um ponto de interrogação forma um prefixo, que deve especificar um documento. O servidor usa suas informações de configuração para determinar como mapear o prefixo para um programa CGI em particular. Quando um servidor invoca o programa CGI, ele passa um argumento ao programa que consiste em todos os caracteres na URL após o ponto de interrogação.

Porque o padrão CGI foi originalmente projetado para servidores Web que executam sobre sistemas operacionais como UNIX e Windows, ele segue uma convenção incomum ao passar argumentos a programas CGI. Em vez de usar a linha de comando, um servidor coloca informações sobre argumentos em *variáveis de ambiente* UNIX e então invoca o programa CGI. O programa CGI herda uma cópia das variáveis de ambiente, da qual extrai os valores. Por exemplo, o servidor designa o sufixo da URL à variável de ambiente *QUERY_STRING*².

Um servidor designa também valores a outras variáveis de ambiente que o programa CGI pode usar. A tabela na Figura 36.2 lista exemplos de variáveis de ambiente CGI junto com seus significados.

36.9 Informações de estado e cookies

Um servidor invoca um programa CGI cada vez que uma requisição chega para uma URL associada. Além disso, por não manter qualquer histórico de requisições, o servidor não pode dizer ao programa CGI sobre requisições anteriores oriundas do mesmo usuário. Não obstante, um histórico é útil porque permite que um programa CGI participe de um diálogo. Por exemplo, um histórico de interações anteriores possibilita evitar que um usuário responda perguntas repetidas ao especificar requisições adicionais. Para fornecer um diálogo não repetitivo, um programa CGI deve salvar informações entre requisições.

Nome da variável	Significado
SERVER_NAME	O nome de domínio do computador que está executando o servidor.
GATEWAY_INTERFACE	A versão do software CGI que o servidor está utilizando.
SCRIPT_NAME	O caminho na URL depois do nome do servidor.
QUERY_STRING	Informações após "?" na URL.
REMOTE_ADDR	O endereço IP do computador que está executando o navegador que enviou a requisição.

Figura 36.2 Exemplos de variáveis de ambiente passadas a um programa CGI.

² Na realidade, o servidor codifica o sufixo usando o formato de URL padrão: cada espaço em branco é substituído por um sinal de adição (+), e caracteres não imprimíveis são substituídos por um caractere de porcento (%) seguido por dois dígitos hexadecimais que fornecem o valor numérico do caractere especial.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

Tecnologia para Documentos Web Ativos (Java e JavaScript)

37.1 Introdução

Os dois capítulos anteriores introduzem a World Wide Web e descrevem os três tipos de documentos Web. O capítulo anterior explica como as tecnologias do tipo CGI permitem que um servidor gere páginas dinâmicas.

Este capítulo considera a terceira forma de documentos Web: ativos. Ele descreve a motivação para documentos ativos, explica os conceitos básicos e compara a abordagem de documento ativo com uma tecnologia anterior baseada em servidor e usada para atualização contínua. Além disso, o capítulo descreve os passos de tradução usados para preparar e carregar um documento ativo. Finalmente, o capítulo considera a tecnologia de Java usada para criar e executar documentos ativos; ele caracteriza a linguagem Java e apresenta um exemplo simples de um *applet* Java.

37.2 Atualização contínua com server push e client pull

À medida que o HTTP e os navegadores Web foram sendo projetados, se tornou aparente que documentos dinâmicos não bastariam para todos os propósitos. Por as informações contidas em um documento dinâmico serem fixadas quando ele é criado, um documento dinâmico não pode atualizar informações na tela à medida que informações mudam, nem mudar imagens gráficas para fornecer animação.

Duas técnicas mais antigas foram inventadas para permitir a atualização contínua da tela de um usuário: *server push* e *client pull*. O *server push* exige que um servidor periodicamente gere e envie novas cópias de um documento; *client pull* requer que o navegador faça pedidos contínuos e exiba versões sucessivas de um documento. Em ambos os casos, o usuário vê os conteúdos da página mudarem continuamente enquanto a página é visualizada.

Embora as técnicas antigas permitissem que os documentos mudassem, nenhuma funcionava bem. Causavam excessiva sobrecarga ao servidor e não suportavam atualizações rápidas. A sobrecarga é especialmente significativa porque um servidor deve satisfazer a muitos clientes simultaneamente e, às vezes, adaptar o conteúdo para cada usuário (ou seja, deve executar um programa aplicativo para cada cópia da página). As técnicas não permitem mudanças rápidas, pois não funcionam bem em larga escala: quando muitos navegadores requerem mudanças contínuas, a

carga no servidor aumenta, o que significa que este leva mais tempo para gerar e enviar a página. Para resumir:

As primeiras técnicas para fornecer atualização contínua de informações em uma página Web são o server push e o client pull. Nenhuma funcionava bem.

37.3 Documentos ativos e sobrecarga do servidor

Os *documentos ativos* surgiram como uma técnica alternativa para fornecer atualização contínua sem sobrecarregar os servidores. A abordagem de documento ativo move o processamento para o navegador; um navegador recebe um programa de computador que o navegador executa localmente. Uma vez que envia uma cópia do documento ativo, o servidor não tem mais nenhuma responsabilidade com a execução ou atualização. Deste modo, diferentemente das abordagens anteriores, a abordagem de documento ativo não usa a CPU do servidor. Além disso, um documento ativo pode atualizar a tela de exibição do usuário sem usar largura de banda.

37.4 Representação de documento e tradução ativas

Que representação deve ser usada para um documento ativo? A resposta é complexa. Porque um documento ativo é um programa que especifica como processar e exibir informações, muitas representações são possíveis. Os programadores preferem a representação textual. Uma representação interna, geralmente binária, é utilizada para alcançar altas velocidades de execução. Finalmente, uma representação comprimida pode ser usada para minimizar o atraso e a largura de banda de comunicação necessária para transferir um programa através da Internet.

Para satisfazer todos os requisitos para representação, tecnologias de documento ativo usam a mesma abordagem geral para a representação de programas usada por uma linguagem de programação convencional: um compilador e um linker são usados para traduzir automaticamente entre as representações. A Figura 37.1 mostra os passos seguidos.

Embora algumas tecnologias de documento ativo façam com que um navegador aceite e interprete um documento fonte, a maioria dos sistemas fornece uma otimização para documentos maiores: um *compilador* é usado para traduzir um documento da representação fonte para uma *repre-*

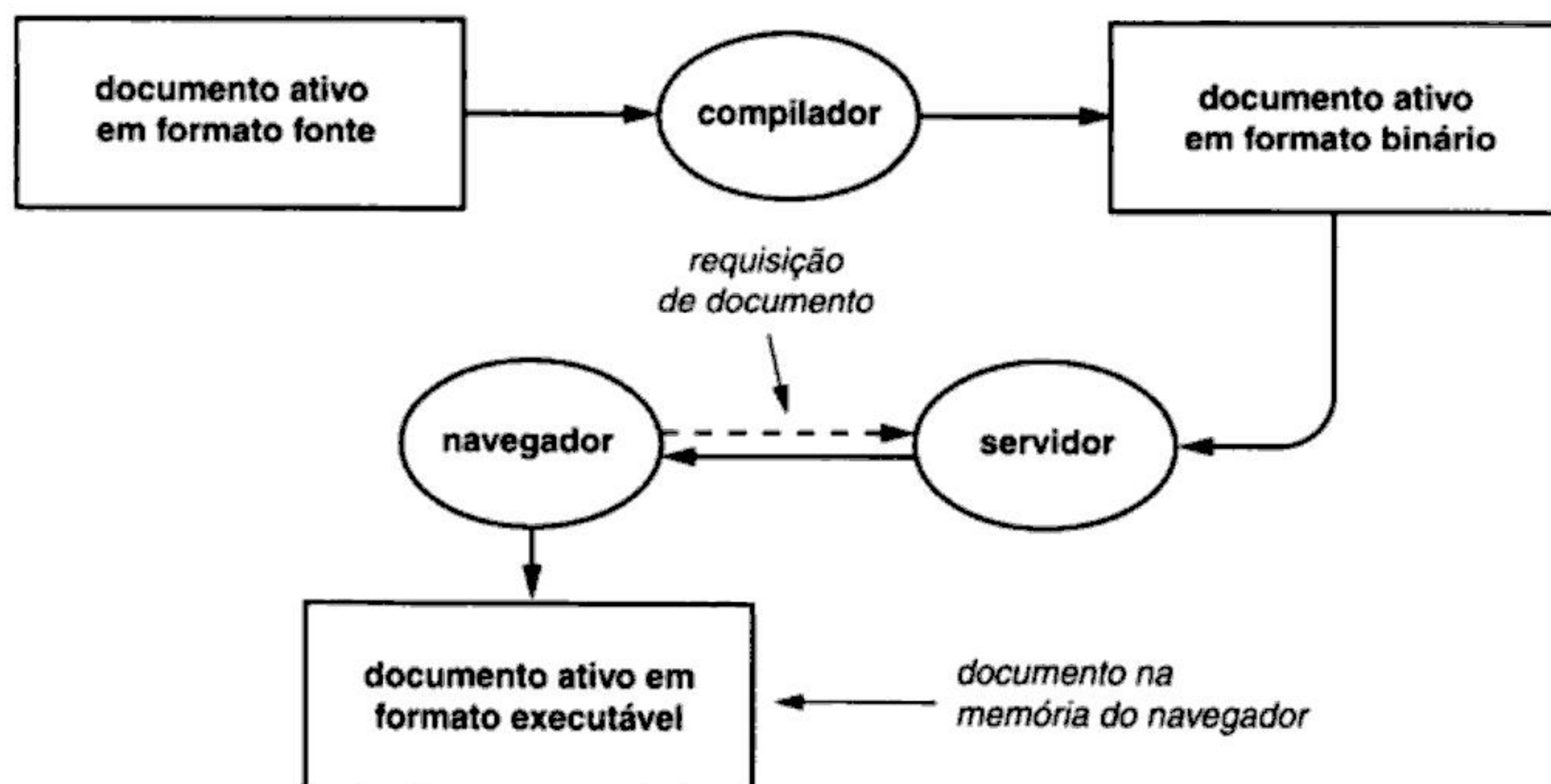


Figura 37.1 Ilustração de três representações de documento ativo e os programas que traduzem ou transportam o documento. As setas escurecidas mostram a direção na qual um documento se move.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

- *E/S de rede de baixo nível.* A biblioteca contém classes que fornecem acesso às facilidades de E/S em nível de sockets no sistema de execução. Um programa Java pode usar facilidades de E/S de rede de baixo nível para enviar e receber datagramas UDP, ou abrir uma conexão TCP para enviar e receber streams de dados.
- *Interação com um servidor Web.* Um applet pode precisar acessar documentos Web estáticos ou dinâmicos ou outros applets. Por exemplo, um applet pode seguir uma URL para carregar e mostrar um documento estático (como uma imagem). A biblioteca contém classes para tratar de tais tarefas.
- *Acesso ao sistema de execução.* A biblioteca Java contém classes que um applet pode usar para acessar facilidades no ambiente de execução. Por exemplo, um programa em execução pode solicitar a criação de uma thread.
- *E/S de arquivo.* Um applet usa facilidades de E/S de arquivo para manipular arquivos no computador local. A E/S de arquivo é usada por programas para salvar informações de estado de longo prazo.
- *Estruturas de dados convencionais.* A biblioteca inclui classes que definem estruturas de dados convencionais. Por exemplo, um programa pode usar uma classe *dicionário* para criar uma estrutura de dados eficiente que armazene itens para recuperação posterior.
- *Captura de eventos.* Eventos acontecem quando um usuário solta um botão do mouse ou tecla no teclado. A biblioteca contém classes que permitem que um programa Java capture e trate de tais eventos.
- *Tratamento de exceções.* Quando uma condição inesperada ou um erro acontece em um programa Java, o ambiente de execução provoca uma *exceção*. A biblioteca Java inclui um conjunto de classes que tornam o tratamento de exceções mais fácil.

Para resumir:

Java é uma tecnologia usada para criar e executar documentos ativos, chamados de applets. A tecnologia Java inclui uma nova linguagem de programação, uma extensa biblioteca de classes e um ambiente de execução.

37.9 Um conjunto de ferramentas gráficas

O ambiente de execução Java inclui facilidades que permitem que um applet manipule a tela de um usuário, e a biblioteca Java contém software que fornece uma interface gráfica de alto nível. Juntos, o suporte gráfico do sistema de execução e a biblioteca gráfica são conhecidos como um conjunto de ferramentas gráficas; o conjunto de ferramentas gráficas específico do Java é conhecido como *Abstract Window Toolkit (AWT)*⁴. Existem duas razões pelas quais o Java necessita de um conjunto de ferramentas gráficas significativo. Primeiro, o propósito central de um applet é uma apresentação complexa – um applet é usado sempre que uma tela estática é inadequada. Segundo, um programa que controla uma tela gráfica deve especificar muitos detalhes. Por exemplo, quando um programa Java precisa mostrar um novo item ao usuário, ele deve escolher entre mostrar o item em um sub-área de uma janela existente ou criar uma nova janela independente. Se uma nova janela é criada, o programa deve especificar um cabeçalho para ela, o tamanho da janela, as cores a serem usadas e detalhes como onde colocar a janela e se ela tem uma barra de rolagem.

⁴ Em vários momentos, os projetistas expandiram AWT também para *Alternative Window Toolkit*, *Advanced Window Toolkit* e *Applet Widget Toolkit*.

O AWT não especifica um estilo gráfico particular ou nível de interação. Em vez disso, o conjunto de ferramentas inclui classes para manipulação de baixo e alto níveis; um programador pode escolher administrar os detalhes ou invocar métodos do conjunto de ferramentas para fazê-lo. Por exemplo, o conjunto de ferramentas não dita a forma de uma janela. Por um lado, o conjunto inclui classes de baixo nível para criar um painel retangular branco na tela, desenhar linhas ou polígonos, ou ainda capturar teclas e eventos de mouse que acontecem no painel. Por outro lado, ele inclui classes de alto nível que fornecem uma janela completa com um cabeçalho, bordas e barras de rolagem verticais e horizontais. Um programador pode escolher usar o estilo de janela fornecido ou programar todos os detalhes.

De forma semelhante, um programador deve escolher entre facilidades de alto e de baixo níveis para interação com o usuário. Por exemplo, um applet pode usar classes de alto nível do conjunto de ferramentas para criar botões, menus pull-down ou quadros de diálogos na tela. Alternativamente, um applet pode criar tais itens usando classes do conjunto de ferramentas de baixo nível para desenhar linhas, especificar matização e controlar a fonte usada para exibir texto.

Porque um applet pode precisar interagir com documentos estáticos, o conjunto de ferramentas inclui classes que executam operações convencionais de navegação Web. Por exemplo, dada uma URL, um applet pode usar classes do conjunto de ferramentas para buscar e mostrar um documento HTML estático ou uma imagem, ou ainda buscar e tocar um clip de áudio.

Para resumir:

Java inclui um extenso conjunto de ferramentas gráficas que consiste em suporte de execução para imagens gráficas, bem como software de interface na biblioteca. O conjunto de ferramentas permite que um programador escolha uma interface de alto nível, cujos detalhes são tratados pelo conjunto de ferramentas, ou uma interface de baixo nível, em que o applet trata dos detalhes.

37.10 Usando Java gráfico em um computador em particular

Porque a Abstract Window Toolkit foi projetada para ser independente de hardware de computador e sistema gráfico de um vendedor específico, o Java não precisa usar diretamente hardware de tela. Ele pode executar sobre sistemas de janela convencionais. Por exemplo, se um usuário executa um navegador em um computador que usa o X Window System, um applet Java usa *X* para criar e manipular janelas na tela. Se outro usuário executa um navegador com um sistema de janelas diferente, um applet usará aquele sistema de janelas para criar e manipular a tela.

Como um applet Java pode funcionar com múltiplos sistemas de janela? A resposta reside em um mapeamento importante construído no ambiente de execução e um conjunto de funções intermediárias. Um ambiente de execução Java inclui uma camada intermediária de software. Cada função nessa camada usa o sistema de janelas do computador para implementar uma operação gráfica específica de Java. Antes de um applet iniciar, o ambiente de execução estabelece um mapeamento entre cada método gráfico de Java e a função intermediária apropriada. Quando um applet solicita uma operação com AWT, o controle passa para um método na biblioteca Java. O método da biblioteca então encaminha o controle para a função intermediária apropriada, que usa o sistema de janelas do computador para fornecer a operação. O conceito é conhecido como *fábrica (factory)*.

As vantagens de usar uma camada intermediária de software são portabilidade e generalidade. Porque funções intermediárias usam o sistema de janelas do computador, somente uma versão da biblioteca AWT é necessária – a biblioteca não contém código relacionado ao hardware gráfico específico. Porque a amarração entre métodos gráficos do Java e o software do sistema de janelas do computador é estabelecida pelo sistema de execução, um applet Java é portátil. A desvantagem principal é a imprecisão que surge por a camada intermediária de software poder interpretar ope-



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

A Figura 37.3 mostra como a tela aparece depois de *init* acrescentar dois itens gráficos à janela do applet.

O coração do applet exemplo é o método *action*. Se o botão marcado com *Click Here* for selecionado, o applet incrementa a variável *count* e muda a tela. Para fazê-lo, o applet invoca o método *setText*, que substitui a mensagem em *f*. Deste modo, cada vez que um usuário clica no botão, a tela muda. Por exemplo, a Figura 37.4 mostra como a tela aparece depois de o usuário clicar o botão uma vez.

Do ponto de vista de um usuário, o applet exemplo parece operar de forma semelhante ao script CGI na Figura 36.4⁵. Do um ponto de vista da implementação, a duas operam de maneiras bem diversas. Diferentemente de um script CGI, um applet mantém informações de estado localmente. Deste modo, diferentemente do script CGI exemplo, o applet exemplo não contata um servidor antes de atualizar a tela.

37.14 Invocando um applet

Dois métodos podem ser usados para invocar um applet. Primeiro, um usuário pode prover a URL de um applet para um navegador que entenda Java. Quando o navegador contactar o servidor especificado na URL, o servidor informará ao navegador que o documento é um applet Java. Segundo, um documento HTML pode conter uma *applet tag* que se refira a um applet. Quando um navegador encontrar o marcador, ele contactará o servidor para obter uma cópia do applet.

Em sua forma mais simples, uma *applet tag* especifica a localização de um arquivo *.class* a ser buscado e executado. Por exemplo, suponha que o servidor Web na máquina *www.nonexist.edu* armazene o arquivo *bbb.class* no diretório *example*. A URL para o applet é:

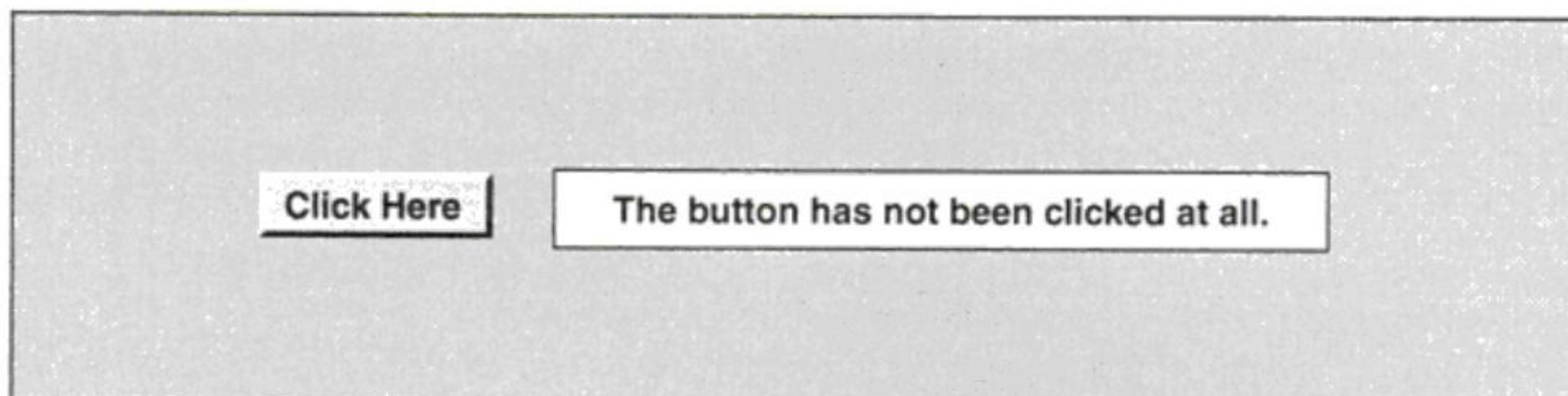


Figura 37.3 Ilustração da tela após o applet na Figura 37.2 iniciar a execução.

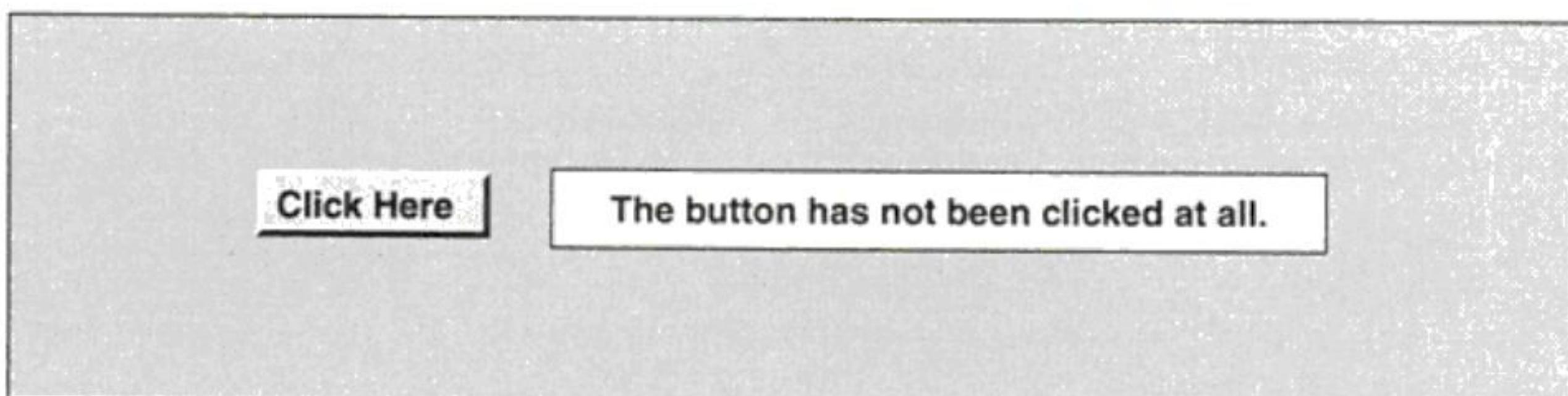


Figura 37.4 A tela depois de o usuário clicar o botão uma vez.

⁵ A Figura 36.4 pode ser encontrada na página 503.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

Sumário do Capítulo

- 41.1** Introdução 561
- 41.2** Software de Protocolo Bootstrapping 561
- 41.3** Parâmetros de Protocolo 562
- 41.4** Configuração de Protocolo 562
- 41.5** Exemplos de Itens que Precisam ser Configurados 562
- 41.6** Configuração a partir de Armazenamento Estável 563
- 41.7** A Necessidade de Automatizar a Configuração de Protocolo 563
- 41.8** Métodos para Configuração Automatizada de Protocolo 563
- 41.9** O Endereço Usado para Encontrar um Endereço 564
- 41.10** Uma Seqüência de Protocolos Usados Durante o Bootstrap 565
- 41.11** Protocolo Bootstrap (BOOTP) 565
- 41.12** Dynamic Host Configuration Protocol (DHCP) 567
- 41.13** Otimizações em DHCP 568
- 41.14** Acesso Indireto ao Servidor Através de um Relay 568
- 41.15** Formato da Mensagem de DHCP 568
- 41.16** DHCP e Nomes de Domínio 569
- 41.17** Resumo 570

Inicialização (Configuração)

41.1 Introdução

Os capítulos anteriores consideram como redes e inter-redes operam. Eles explicam como um software de protocolo permite a um computador se comunicar com outros computadores e como pacotes fluem através de uma inter-rede. Capítulos posteriores descrevem o paradigma cliente-servidor e explicam como aplicativos usam a interação cliente-servidor quando eles se comunicam. Em todos os casos, o livro assume que os sistemas dos hosts e roteadores já estão executando. Ou seja, cada computador já foi ligado, o sistema operacional iniciou, o software de protocolo foi carregado e valores como entradas na tabela de roteamento já foram inicializados.

Este capítulo considera uma questão fundamental: como o software de protocolo em um host ou roteador começa a operar? Em particular, que passos um sistema de computador deve tomar antes de o software de protocolo estar pronto para uso? Este capítulo descreve as variáveis que devem ser inicializadas e explica mecanismos que podem ser usados para designar valores àquelas variáveis. O capítulo explica por que uma inicialização automática é necessária e descreve um protocolo que computadores podem usar para obter informações automaticamente. Surpreendentemente, será visto que inicialização é outro exemplo de programas aplicativos que usam cliente-servidor.

41.2 Software de protocolo bootstrapping

O que acontece quando um computador começa a operar? O processo é conhecido como *bootstrapping*¹. O hardware carrega um *programa de boot* (*boot program*) para a memória, que por sua vez inicia um programa maior, geralmente um sistema operacional. O sistema operacional contém o software de protocolo, que precisa ser inicializado antes que possa ser utilizado.

¹ Às vezes abreviado de *booting*, o termo deriva da frase em inglês “pulling oneself up by one's own bootstraps” (N. do T.: esta frase é uma expressão idiomática cujo significado é aproximadamente “tentar melhorar sua posição sem ajuda, através de seu próprio esforço”.)

41.3 Parâmetros de protocolo

O software de protocolo precisa ser informado sobre o computador local e o ambiente de rede local. Por exemplo, o software deve saber o tipo de hardware de rede (por exemplo, se é uma Ethernet), os protocolos a serem usados (por exemplo, os protocolos TCP/IP), e a localização de serviços tais como o DNS. Para tornar o software de protocolo geral e portátil, os programadores não fixam todos os detalhes no código fonte. Em vez disso, eles *parametrizam* o software de protocolo para possibilitar o uso de uma única imagem binária em diversos computadores e o uso de uma única imagem binária em um único computador no qual as conexões de Internet mudam com o tempo. Cada detalhe que difere de um computador para outro ou de uma rede para outra é codificado em um parâmetro separado que pode ser mudado. Conseqüentemente, antes que o software possa ser executado, um valor deve ser fornecido para cada parâmetro.

Para resumir:

O software de protocolo é parametrizado para permitir que um binário compilado execute em múltiplos computadores e em uma variedade de ambientes de rede sem modificações. Quando uma cópia do software é iniciada em um determinado computador, informações sobre o computador e a rede devem ser fornecidas para o software através de um conjunto de parâmetros.

41.4 Configuração de protocolo

O ato de fornecer valores para parâmetros no software de protocolo é conhecido como *configuração de protocolo*. Depois de um valor ter sido fornecido para cada parâmetro, diz-se que o software de protocolo está *configurado*. O software de protocolo deve ser configurado antes dele poder ser usado.

Embora o conceito de configuração de protocolo seja fácil de entender, a implementação pode ser complicada por três razões. Primeiro, existem diversos mecanismos para obter as informações necessárias. Segundo, existem diversos métodos para passar as informações ao software de protocolo. Terceiro, um determinado sistema de computador pode decidir usar múltiplos métodos – alguns parâmetros podem ser obtidos e especificados usando um método, enquanto outros parâmetros são obtidos ou especificados usando outros métodos. Além disso, o sistema pode permitir que algumas peças do software de protocolo sejam usadas antes que todos os pedaços estejam configurados. As próximas seções discutem sobre configuração e mostram que pode ser útil permitir que protocolos operem após uma configuração parcial.

41.5 Exemplos de itens que precisam ser configurados

As informações de configuração que o software de protocolo precisa podem ser divididas em duas classes gerais: internas e externas. Informações internas estão relacionadas ao computador propriamente dito (por exemplo, o endereço de protocolo do computador). Informações externas pertencem ao ambiente que cerca um computador (por exemplo, a localização das impressoras que podem ser alcançadas através da rede).

Os detalhes exatos das informações de configuração dependem da pilha de protocolos. Por exemplo, os itens que o software de protocolo do TCP/IP precisa incluem:

- *Endereço(s) IP.* Cada computador deve ter um endereço IP único para cada interface.
- *Endereço de roteador default.* O software de IP deve saber o endereço de um roteador que pode ser usado para alcançar lugares remotos; o endereço se torna o próximo hop para uma rota default na tabela de roteamento do computador.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

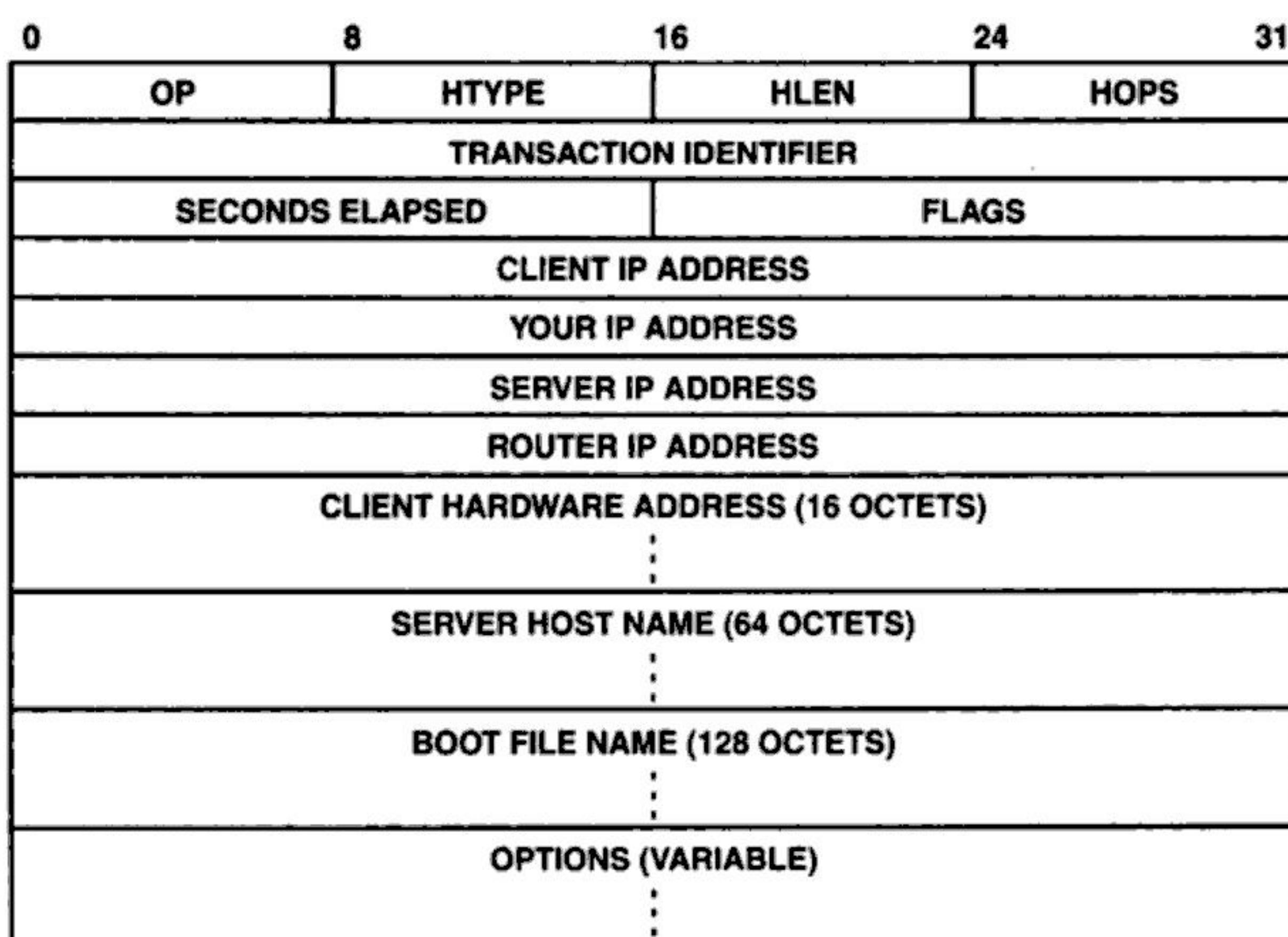


Figura 41.3 O formato de mensagem do DHCP, uma versão ligeiramente modificada do formato do BOOTP.

A maioria dos campos em uma mensagem de DHCP tem o mesmo significado que em *BOOTP*; DHCP substitui o campo *UNUSED* de 16 bits com um campo *FLAGS*, e usa o campo *OPTIONS* para codificar informações adicionais. Por exemplo, como em *BOOTP*, o campo *OP* especifica ou uma *requisição* ou uma *resposta*. Para distinguir entre várias mensagens que um cliente usa para descobrir servidores ou solicitar um endereço, ou que um servidor usa para confirmar ou negar uma requisição, o DHCP usa uma *opção de tipo de mensagem*. Isto é, cada mensagem contém um código que identifica o tipo da mensagem.

41.16 DHCP e nomes de domínio

Embora DHCP possibilite que um computador obtenha um endereço IP sem intervenção manual, o DHCP não interage com o Domain Name System. Como resultado, um computador não pode manter seu nome quando ele mudar de endereço. É interessante que o computador não precisa mover para uma nova rede para ter que mudar de nome. Por exemplo, suponha que um computador obtenha o endereço IP *192.5.48.195* a partir do DHCP, e suponha que o sistema de nomes de domínios contenha um registro que amarra o nome *x.y.z.com* ao endereço. Agora considere o que acontece se o proprietário desliga o computador e sai de férias por dois meses, durante os quais o empréstimo de endereço expira. O DHCP pode designar o endereço a outro computador. Quando o proprietário retorna e liga o computador, DHCP negará a requisição para usar o mesmo endereço. Desta modo, o computador obterá um novo endereço. Infelizmente, o DNS continua a mapear o nome *x.y.z.com* para o endereço antigo⁷.

⁷ Algumas implementações do DHCP usam um mecanismo de propriedades para atualizar o servidor DNS local sempre que um endereço for atribuído.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

Glossário de Termos e Abreviações de Ligação em Redes

Terminologia de ligação em redes

A terminologia em rede pode ser extremamente confusa para os iniciantes porque não é lógica nem consistente. Existem três razões para isso. Primeiro, como não existe uma única teoria que explique tudo em redes, a terminologia não é derivada de um framework teórico subjacente. Segundo, como muitos grupos e organizações desenvolveram e padronizaram tecnologias em rede, existem variações na terminologia. Terceiro, os praticantes inventaram termos informais e abreviações que são freqüentemente utilizados no lugar de termos técnicos formais.

Este glossário inclui a terminologia usada na prática e termos técnicos encontrados na literatura. As definições são breves – não incluem explicações detalhadas de termos nem exemplos – porque têm o objetivo de refrescar a memória do leitor fornecendo uma visão global do conceito geral associado com cada termo.

As siglas e os termos são listados em ordem alfabética, com a expansão de uma sigla entre parênteses. Por exemplo, a sigla *MTU* é seguida pela expansão (*Maximum Transmission Unit*).

Termos e abreviações de redes em ordem alfabética

10/100 Ethernet

Uma interface Ethernet que pode detectar automaticamente a máxima velocidade de um dispositivo conectado e escolher operar a 10 Mbps ou 100 Mbps.

10Base-T

O nome técnico para Ethernet de par trançado de 10 Mbps.

100Base-T

O nome técnico para Ethernet rápida de 100 Mbps.

1000Base-T

O nome técnico para Ethernet de gigabit.

2-3 swap (troca 2-3)

Uma referência para o cabo no qual o fio usado para transmitir, em uma ponta, conecta com o fio para receber na outra ponta, e vice-versa. Os números 2 e 3 se referem aos pinos de transmissão e recepção em um conector DB-25.

2B+D service (serviço 2B+D)

Um serviço ISDN que inclui duas conexões de telefone padrão mais uma conexão de dados.

3-way handshake (handshake de 3 modos)

Uma técnica usada pelo TCP e outros protocolos de transporte para, de forma confiável, começar ou terminar uma comunicação.

3-wire circuit (círculo de 3 fios)

Um esquema de fiação freqüentemente utilizado com conexões seriais assíncronas entre um par de computadores. O primeiro fio é usado para transmitir dados de um computador para outro, o segundo é usado para dados em uma direção reversa e o terceiro é um fio comum de terra.

4-wire circuit (círculo de 4 fios)

Um esquema de fiação freqüentemente utilizado com conexões seriais assíncronas entre um par de computadores. Um par de fios é usado para transportar dados em uma direção e outro par para transportar dados na direção reversa. Um 4-wire circuit é usado mais para longas distâncias do que o 3-wire circuit.

5-layer reference model

Um modelo conceitual que explica como os protocolos Internet TCP/IP são projetados.

7-layer reference model

Um modelo conceitual mais novo, difundido pela ISO (International Organization for Standardization), para especificar como um conjunto de protocolos trabalha para fornecer serviços de comunicação. O "7-layer reference model" não inclui a camada do protocolo da Internet.

802.2

O padrão IEEE para Logical Link Control. Ver LLC e SNAP.

802.3

O padrão IEEE para Ethernet.

802.5

O padrão IEEE para Token Ring.

802.11b

O padrão IEEE para uma rede sem fio wi-fi.

802.1x

O padrão IEEE para um mecanismo de segurança usado com rede sem fio.

AAL5 (ATM Adaptation Layer 5)

O protocolo usa uma aplicação para enviar dados sobre uma rede ATM. AAL5, o qual é parte da pilha ATM, aceita e entrega pacotes grades de dados (acima de 64 bytes).

ABR (Taxa de Bits Disponível)

Um tipo de serviço oferecido pela ATM.

access delay

O tempo que uma interface da rede espera antes de acessar uma rede compartilhada.

ACL (Access Control List)

Um mecanismo de segurança que especifica usuários e computadores que podem ter acesso a um objeto particular.

broadband technology (tecnologia de banda larga)

O termo utilizado para descrever uma tecnologia de rede que usa uma grande parte do espectro eletrônico para obter taxas mais altas de throughput. Normalmente, os sistemas de banda larga empregam multiplexação por divisão de freqüência para permitir que múltiplas comunicações independentes prossigam simultaneamente através de um único meio subjacente. Ver tecnologia de banda base (baseband).

broadcast

Uma forma de transmissão na qual uma cópia de um pacote é entregue a cada computador em uma rede. Ver cluster, multicast e unicast.

broadcast address (endereço de broadcast)

Um endereço especial que faz com que o sistema adjacente entregue uma cópia de um pacote para todos os computadores na rede.

broadcast direto

Um broadcast para todos os computadores de uma rede remota alcançada pelo envio de uma cópia simples do pacote para a rede remota e reenviando o pacote quando ele chega. O TCP/IP suporta broadcast direto.

broadcast satellite (broadcast por satélite)

Um sistema de rede no qual um satélite transmite todos os pacotes e permite que um receptor individual descarte pacotes direcionados para outros receptores.

byte stuffing (recheio de byte)

Uma técnica de protocolo em que os dados são alterados inserindo-se bytes adicionais para distinguir entre valores de dados e campos de controle de pacotes.

cabeçalho base (base header)

O cabeçalho necessário encontrado no início de um datagrama do IPv6.

cabeçalho de extensão

Um cabeçalho opcional usado no protocolo de IPv6.

cable modem (modem de cabo)

Um modem utilizado para enviar informações digitais através de cabos coaxiais usados pelas televisões a cabo.

cabo de categoria 5 (Category 5 cable)

Um tipo de cabeamento necessário para Ethernet de par trançado. As características elétricas de cabo de categoria 5 tornam-o menos suscetível às interferências elétricas do que cabos de categorias mais baixas.

carrier signal (sinal da portadora)

O sinal básico transmitido através das redes. Uma portadora é modulada (isto é, mudado) para codificar dados.

CAT (Cable Address Translation)

Uma forma de NAT(Net Address Translation) proposta pelas companhias de TV a cabo.

category 5 cable (cabo de categoria 5)

Um tipo de cabeamento necessário para Ethernet de par trançado. As características elétricas de cabo de categoria 5 tornam-o menos suscetível às interferências elétricas do que cabos de categorias mais baixas.

CATV (Community Antenna TeleVision)

O nome usado pelos sistemas de televisão a cabo. A tecnologia CATV usa multiplexação de divisão de freqüência para propagar simultaneamente múltiplos canais de televisão sobre um simples cabo.

CBR (Constant Bit Rate)

O serviço usado pela ATM para transmitir uma seqüência de áudio ou vídeo. Dados são enviados em uma taxa sem variação.

CBT (Core Base Trees)

Um protocolo para propagar roteamento multicast de informação.

CCITT (Consultative Committee on International Telephone and Telegraph)

O nome antigo da ITU.

célula (cell)

Um pequeno pacote de tamanho fixo (isto é, as redes ATM enviam células de 48 bytes)

cell tax

Um termo pejorativo usado para referenciar o sobre peso de 10% do cabeçalho introduzido pela ATM.

CGI (Common Gateway Interface)

Uma tecnologia usada para criar documentos dinâmicos da WWW. Programas CGI rodam no servidor.

chat

Um serviço da Internet que permite múltiplos usuários comunicarem-se através de texto enviado por um usuário para as telas dos outros participantes.

checksum

Um valor usado para verificar que os dados não foram corrompidos durante a transmissão. O computador que envia calcula um checksum adicionando valores binários dos dados e transmite o resultado no pacote com os dados. O computador que recebe recalcula o checksum e compara com o recebido no pacote. Ver CRC.

cabo coaxial

Um tipo de cabo utilizado para redes de computadores e para televisões a cabo. O nome deriva da estrutura em que uma proteção de metal cerca um fio central. A proteção protege o sinal no fio interno de interferência elétrica.

canal B (bearer channel)

O termo que as companhias telefônicas usam para representar um canal configurado para tratar de um circuito telefônico de voz. O ISDN inclui serviço de canal B. Ver canal D.

canal D

O termo que as companhias telefônicas usam para representar um canal configurado para tratar de dados. O ISDN inclui um serviço de canal D. Ver canal B.

canal virtual

Um sinônimo para circuito virtual. O termo canal virtual é usado por tecnologias como ATM. Ver VC.

carga de dados

Genericamente, os dados sendo transportados em um pacote. A carga de dados de um quadro são os dados no quadro; a carga de dados de um datagrama é a área de dados do datagrama.

CIDR (Classless Inter-Domain Routing)

O esquema de roteamento e endereçamento do IP que substitui o endereçamento completo. CIDR usa uma máscara de endereço de 32 bits para marcar o limite entre o prefixo e o sufixo do endereço IP.

circuit switching

Uma alternativa para produzir switches de pacotes no qual o hardware de rede forma um circuito entre dois endpoints. A rede original de telefones usou circuit switching.

CL (ConnectionLess)

Característica de uma rede na qual cada transmissão é independente. A alternativa para rede sem conexão é rede orientada à conexão. Ver orientada à conexão.

codificação Manchester (Manchester encoding)

A codificação da camada física que a Ethernet usa para transmitir um frame.

ColdFusion

Uma tecnologia de página Web dinâmica que permite carregar itens de uma base de dados convencional. O servidor interpreta consultas SQL aninhadas nas páginas sempre que elas são requisitadas.

colisão

Um evento que acontece em uma rede CSMA/CD quando duas estações tentam transmitir simultaneamente. Os sinais interferem um com o outro, forçando as duas estações a recuar e tentar novamente.

compressão zero

Uma técnica que o IPv6 utiliza para abreviar a notação hexadecimal de dois pontos substituindo uma seqüência de zeros por um par de caracteres de dois pontos.

comutação (switching)

Um termo geral usado para descrever a operação de um switch. Como é associada ao hardware, a comutação é usualmente mais rápida que o roteamento. Além disso, a comutação difere do roteamento porque a comutação utiliza o endereço de hardware em um quadro.

comutação de rótulo (label switching)

Uma técnica usada para comutar tecnologias tal como ATM na qual cada switch que manipula uma célula reescreve o rótulo da célula.

comutação IP (IP switching)

Um esforço para combinar IP com hardware de comutação (especialmente ATM) para alcançar alta velocidade. Ver comutação na camada 3.

comutação na camada 3 (layer-3 switching)

Um esforço para combinar IP com hardware de comutação (especialmente ATM) para alcançar alta velocidade. Ver comutação IP.

conector BNC

O tipo de conector usado com Ethernet de fio fino.

configuração de protocolo

Um passo que um sistema de computador deve executar para atribuir valores a parâmetros antes que o software de protocolo possa ser usado. Normalmente, a configuração de protocolo exige que um sistema obtenha um endereço de protocolo.

confirmação (acknowledgment)

Uma pequena mensagem retornada para informar a um remetente que os dados chegaram em seu destino pretendido.

congestionamento

Uma condição em que cada pacote enviado através de uma rede sofre atraso excessivo porque a rede é atropelada com pacotes. A menos que o software de protocolo detecte congestionamento e reduza a taxa em que os pacotes são enviados, uma rede pode sofrer colapso de congestionamento.

contagem de hop (hop count)

Um número em um cabeçalho de protocolo que determina quantas máquinas intermediárias um pacote visita. Protocolos como IP exigem que o remetente especifique uma contagem máxima de hops; isto impede que um pacote viaje ao redor de um loop de roteamento para sempre.

controle de fluxo

Um mecanismo de protocolo que permite que um receptor controle a taxa em que um remetente transmita dados. O controle de fluxo possibilita que um receptor sendo executado em um compu-

tador de baixa velocidade aceite dados de um computador de alta velocidade sem ser atropelado (*overrun*).

connectionless (sem conexão)

Uma característica de um sistema de rede que permite um computador enviar dados para qualquer outro computador em qualquer momento. Redes connectionless são análogas a um sistema de correio no qual cada carta leva o endereço do destinatário; cartas podem ser enviadas em qualquer momento. *Ver* orientada à conexão.

cookie de sessão

Um cookie que o navegador conserva na memória. Porque isto é mantido na memória em vez de disco, o cookie de sessão somente persiste enquanto o navegador está rodando.

cookie

Um pequeno valor passado pelo servidor ao browser como resposta para ajudar o servidor identificar o usuário. Um browser(navegador) retorna o cookie quando envia requisições subsequentes do usuário para o servidor que originou o cookie.

CORBA (Common Object Request Broker Architecture)

Uma tecnologia middleware orientada a objeto.

CRC (Cyclic Redundancy Check, ou Teste de Redundância Cílico)

Um valor usado para verificar se os dados não são adulterados durante a transmissão. O remetente computa um CRC e transmite o resultado em um pacote com os dados. Um receptor computa o CRC sobre os dados recebidos e compara o valor com o CRC no pacote. Um CRC é mais complexo de se computar que um checksum, mas pode detectar mais erros de transmissão.

CSMA (Carrier Sense Multiple Access)

A técnica usada com redes de arquitetura de barramento em que os computadores acoplados ao barramento comum verificam a presença de uma portadora antes da transmissão.

CSMA/CD (Carrier Sense Multiple Access with Collision Detection)

Uma rede CSMA que tem a capacidade de detectar erros que resultam quando múltiplas estações transmitem simultaneamente. *Ver* colisão.

datagrama IP

A forma que um pacote é enviado através de uma inter-rede TCP/IP. Cada datagrama tem um cabeçalho que identifica o remetente e o receptor seguido de dados.

DB-25

Um conector de 25 pinos freqüentemente usado com as linhas seriais.

DCE (Data Communication Equipment)

Um termo da companhia telefônica para equipamento de comunicação tais como modem. *Ver* DTE. Note: a abreviação é também usado para Distributed Computing Environment.

demarc

Um termo de companhias telefônicas para o limite entre as facilidades de comunicação de sua propriedade com as de uma organização privada.

demodulador

Um dispositivo que aceita uma onda portadora modulada e extrai as informações usadas para modulá-la. *Ver* modem.

demultiplexar

Um conceito geral que se refere à separação de informações recebidas através de um canal de comunicação comum em seus componentes originais. A demultiplexação acontece em hardware (isto é, sinais elétricos podem ser demultiplexados) e em software (isto é, o software de protocolo pode de demultiplexar mensagens recebidas e passar cada uma ao programa aplicativo correto). *Ver* multiplexar.

DST (Distributed Spanning Tree)

O algoritmo que usa bridges para evitar um ciclo.

DSU / CSU (Data Service Unit / Channel Service Unit)

Um dispositivo eletrônico que conecta um circuito de dados digitais alugado com um computador. O DSU / CSU traduz entre o formato digital usado pelas companhias telefônicas e aquele usado pela indústria de computadores. *Ver modem.*

DTE (Data Terminal Equipment)

Um termo de companhias telefônicas para equipamentos tais como computador que conecta-se à rede.

DTMF (Dual Tone Multi-Frequency)

A codificação para a discagem em tons usada em telephones.

DV (Distance-Vector)

Um dos dois algoritmos básicos usado pelos protocolos de roteamento. A alternativa para o vetor-distância de roteamento é estado de link.

DVMRP (Distance Vector Multicast Routing Protocol)

O protocolo de roteamento multicast usado no MBONE. O DVMRP envia multicast localmente, mas usa tunelamento IP-in-IP para transferir datagramas multicast entre os sites.

DWDM (Dense Wavelength Division Multiplexing)

Uma tecnologia de multiplexação de alta velocidade usada com fibra ótica.

EBCDIC (Extended Binary Coded Decimal Interchange Code)

Um padrão que atribui valores únicos a 256 caracteres, incluindo letras maiúsculas e minúsculas, dígitos, pontuação e caracteres de controle. *Ver ASCII.*

echo reply (resposta de echo)

Uma mensagem usada para teste e depuração. Uma resposta de echo ICMP é retornada em resposta a uma mensagem de requisição de echo ICMP. O programa de ping recebe respostas de echo. *Ver requisição de echo (echo request).*

echo request (requisição de echo)

Uma mensagem usada para teste e depuração. O programa de ping envia mensagens de requisição de echo ICMP para gerar respostas de echo.

EGP (Exterior Gateway Protocol)

Termo aplicado para qualquer protocolo de propagação de rota usado para transferir informações de roteamento entre Sistemas Autônomos. *Ver IGP.*

e-mail (correio eletrônico)

Um aplicativo popular em que um usuário ou computador envia um memorando para um ou mais receptores.

encaminhamento de próximo hop (next-hop forwarding)

A técnica usada por protocolos como IP para encaminhar um pacote ao seu destino final. Embora um determinado roteador não contenha informações completas sobre o caminho que um datagrama seguirá, o roteador conhece o próximo roteador para o qual o datagrama deve ser enviado.

encaminhar (forward)

Ver store and forward.

encapsulamento

A técnica em que as informações a serem enviadas são colocadas dentro da área de dados de um pacote ou quadro. Um pacote de um protocolo pode ser encapsulado em outro (por exemplo, ICMP pode ser encapsulado em IP).

encryption key (chave de cifragem)

O pequeno valor usado ao cifrar dados para se garantir a privacidade. Em alguns esquemas de cifragem, o receptor deve usar a mesma chave para decifrar os dados. Outros esquemas usam um par de chaves - uma para cifrar e uma chave diferente para decifrar.

endereçamento hierárquico

Um esquema de endereçamento em que parte de um endereço fornece informações sobre a localização. Por exemplo, um número telefônico é hierárquico porque começa com um código de área seguido por uma central telefônica.

endereçamento de classes (Classful addressing)

Característica do esquema de endereçamento IP corrente na qual endereços unicast foram divididos dentro de três classes principais. A classe de um endereço determina a divisão entre prefixo e sufixo no endereço. *Ver* endereçamento sem classes (classless addressing).

endereçamento sem classes (Classless addressing)

Característica do esquema de endereçamento IP corrente na qual uma máscara de 32-bits é usada para determinar a divisão entre o prefixo e o sufixo de um endereço IP. *Ver* endereçamento com classes (classful addressing).

endereço

Um valor binário único atribuído ao computador; hardware e software de rede usam o endereço dentro de um pacote para reenviar o pacote para seu destino.

endereço fonte

Um endereço no pacote que especifica o computador que o envia. No frame de hardware o endereço fonte deve ser um endereço de hardware. No datagrama IP o endereço fonte deve ser um endereço IP.

endereço de broadcast (broadcast address)

Um endereço especial que faz com que o sistema subjacente entregue uma cópia de um pacote para todos os computadores em uma rede.

endereço de endpoint

Um termo genérico para qualquer endereço atribuído a um computador e que pode ser usado como um endereço destino. Por exemplo, um endereço IP é um tipo de endereço de endpoint.

endereço de hardware

O endereço atribuído a um computador que está acoplado a uma rede. Um quadro enviado de um computador para outro deve conter o endereço de hardware do receptor. Um endereço de hardware é também chamado de endereço físico ou um endereço MAC.

endereço de Internet

Ver endereço IP.

endereço de loopback

Um endereço especial que é usado para teste ou depuração. Um pacote enviado para o endereço de loopback não é transmitido através de uma rede, mas sim retornado pelo sistema de protocolo como se tivesse chegado através da rede.

endereço de origem

Um endereço em um pacote que especifica o computador que enviou o pacote. Em um quadro de hardware, o endereço de origem deve ser um endereço de hardware. Em um datagrama IP, o endereço de origem deve ser um endereço IP.

endereço de protocolo

Um número atribuído a um computador que é usado como o endereço destino em pacotes enviados para aquele computador. Cada endereço IP tem 32 bits; outras famílias de protocolos usam outros tamanhos de endereços de protocolo.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

multiplexação por divisão de freqüência

Uma técnica de multiplexação geral que permite a múltiplos remetentes transmitirem através de um meio comum. Como cada remetente usa uma freqüência diferente, múltiplos remetentes podem transmitir ao mesmo tempo sem interferência. Ver FDM.

multiplexação por divisão de tempo

Uma técnica geral de multiplexação que permite a múltiplos remetentes transmitir através de um meio comum. Os remetentes alternam o uso do meio.

multiplexar

Um conceito geral que se refere à combinação de fontes independentes de informações em uma forma que pode ser transmitida através de um único canal de comunicação. A multiplexação pode ocorrer em hardware (isto é, sinais elétricos podem ser multiplexados) e em software (isto é, software de protocolo pode aceitar mensagens enviadas por múltiplos programas aplicativos e mandá-las através de uma única rede para destinos diferentes). Ver demultiplexar.

NAPT (Network Address and Port Translation)

A forma mais comum de tradução de endereço na qual os números de portas do protocolo são traduzidos tanto quanto os endereços IP.

NAT (Network Address Translation)

A tecnologia que fornece conectividade para muitos computadores em um site através de um endereço simples em endereço IP válido. NAT reescreve o cabeçalho de cada datagrama que sai e chega para substituir o endereço interno com um endereço IP globalmente válido e vice-versa. Ver NAPT.

navegador (browser)

Um programa de computador que acessa e exibe informações da World Wide Web. Um navegador contém múltiplos programas aplicativos e utiliza o nome de um objeto para determinar qual aplicativo deve ser usado para acessar o objeto. Ver URL.

NBMA (Non Broadcast Multiple Access)

Características de uma tecnologia de que conecta múltiplos computadores mas não suporta broadcast. ATM é um exemplo de uma rede NBMA.

next header

Um campo do cabeçalho do Ipv6 que especifica o tipo do próximo item.

next-hop forwarding

A técnica usada pelos protocolos como IP para avançar um pacote para seu destino. Embora um dado roteador não contenha informação completa sobre o caminho que o datagrama seguirá, o roteador conhece o próximo roteador para o qual o datagram deve seguir.

NFS (Network File System)

Um mecanismo de acesso remoto a arquivos originalmente definido pela Sun Microsystems para uso com o sistema operacional UNIX. O NFS permite que aplicativos em um computador acessem arquivos em um computador remoto.

NIC (Network Interface Card, Placa de Interface de Rede)

Um dispositivo de hardware que se acopla a um computador e conecta o computador a uma rede. Freqüentemente chamado de adaptador de rede.

NIU (Network Interface Unit)

Uma peça do equipamento que uma empresa telefônica fornece para terminar um circuito digital. Ver demarc e NTI.

nó (ou nodo)

Um termo usado para se referir informalmente a um roteador ou computador acoplado a uma rede. O termo é derivado da teoria dos grafos.

notação decimal pontilhada

A notação sintática usada para expressar um endereço IPv4 de 32 bits. Cada octeto é escrito em forma decimal com um ponto separando octetos.

notação slash

Uma forma sintática usada no roteamento e endereçamento no CIDR. Um endereço é dado em decimal pontilhado seguido de slash (barra ao contrário) e um inteiro entre 0 e 32 que representa o tamanho da máscara de endereço.

notação hexadecimal de dois pontos

A notação sintática usada para expressar um endereço IPv6.

número de porta de protocolo

Um pequeno inteiro usado para identificar um programa aplicativo específico em um computador remoto. Os protocolos de transporte como TCP designam a cada serviço um número de porta único (por exemplo, e-mail usa a porta 25).

OC (Optical Carrier, ou Portadora Óptica)

Um conjunto de padrões adotados por portadoras comuns para a transmissão em alta velocidade de informações digitais através de fibra óptica. O OC-1 opera a 51,840 Mbps; OC-*n* executa a *n* vezes aquela taxa de transferência. Ver STS.

OC-3 (Portadora Óptica 3)

Um padrão para codificação de fibra óptica usado nos circuitos digitais de companhias telefônicas populares. Um circuito OC-3 opera a 155,520 Mbps. Ver OC.

orientado à conexão

Uma característica de sistemas de rede que exigem que um par de computadores estabeleça uma conexão antes que os dados sejam remetidos. Redes orientadas à conexão são análogas ao sistema telefônico em que uma chamada deve ser feita e respondida antes de a comunicação iniciar. Ver sem conexão (connectionless).

OSPF (Open Shortest Path First Protocol)

Um protocolo popular usado para propagar informações de roteamento dentro de um simples sistema autônomo.

OUI (Organizationally Unique Identifier)

Um campo em um cabeçalho LLC que especifica qual organização designou os números empregados nas informações de tipo.

pacote

Uma pequena e autocontida porção de dados enviada através de uma rede de computadores. Cada pacote contém um cabeçalho que identifica o remetente e o receptor, bem como os dados a serem entregues.

PAR (Positive Acknowledgement with Retransmission, ou Acknowledgment Positivo com Retransmissão)

A técnica básica que os protocolos empregam para obter entrega confiável. O protocolo receptor retorna um acknowledgment quando chega um pacote. Depois da transmissão de um pacote, o remetente começa um temporizador. Se o acknowledgment não chega antes do temporizador expirar, o remetente retransmite o pacote.

par trançado

Uma forma de cabeamento em que um par de fios é envolto ao redor um do outro repetidamente. Trançar dois fios reduz sua suscetibilidade à interferência elétrica.

provisioned

Um adjetivo usado para descrever uma facilidade de rede de configuração manual (isto é, Um circuito provisioned é estabelecido manualmente). O termo tem origem na indústria de telecomunicações.

PSTN (Public Switched Telephone Network)

Rede de telefones convencionais, incluindo mecanismos para estabelecer a chamada e contabilização de impulsos.

PVC (Permanent Virtual Circuit, ou Circuito Virtual Permanente)

Uma conexão de um computador para outro através de uma rede orientada à conexão. Um PVC é permanente no sentido de que sobrevive à reinicialização de um computador ou falta de energia; um PVC é virtual porque é obtido colocando-se rotas em tabelas de roteamento, e não através da instalação de fios. Ver SVC.

QoS

Ver qualidade de serviço.

quadro

A forma de um pacote que o hardware subjacente aceita e entrega.

queuing delay (atraso de enfileiramento)

Ver atraso de enfileiramento.

RADIUS (Remote Authentication Dial-In User Service)

Uma tecnologia que permite um ISP autenticar clientes.

RARP (Reverse Address Resolution Protocol)

Um protocolo que um sistema de computador usa durante o bootstrap para obter um endereço IP.

reassembly (remontagem)

Ver remontagem.

recusa de serviço (Denial-of-service)

Ver DOS.

rede de comutação de pacotes

Qualquer rede de comunicação que aceita e entrega pacotes de informação individuais. A maioria das redes modernas são de comutação de pacotes.

rede de autocura (self healing network)

Um sistema de rede que tem a habilidade de detectar automaticamente um defeito de hardware e rotear tráfego através de um caminho alternativo. A autocura requer o uso de caminhos redundantes. O FDDI é a tecnologia de rede com autocura mais conhecida.

rede em malha (mesh network)

Uma arquitetura de rede em que um computador tem uma conexão ponto a ponto com outro(s) computador(es). Redes em malha completa (full mesh networks), em que cada par de computadores está diretamente conectado, oferecem o mais alto throughput, mas são incomuns porque são caras e difíceis de mudar. Ver topologia em barramento e topologia em anel.

rede ponto a ponto

Qualquer tecnologia de rede que usa uma tecnologia não-compartilhada para conectar pares de computadores. A tecnologia ponto a ponto é mais popular em Redes de Longo Alcance que em Redes Locais.

rede mesh (mesh network)

Uma arquitetura de rede um computador tem uma conexão ponto a ponto para outro computador. Redes completamente mesh, na qual cada par de computadores está diretamente conectado, oferece alto desempenho mas não são comuns porque elas são caras e difícil de muda-las. Ver topologias de barramento e em anel.

rede switched (switched network)

Qualquer rede que usa switchs em vez de roteadores. Usualmente, comutação implica tecnologia orientada a conexão. A ATM é um exemplo de uma tecnologia que usa comutação.

redirecionar

Uma mensagem de erro do ICMP enviada de um roteador para um host. A mensagem especifica que o host tem uma rota incorreta que deve ser mudada e especifica o destino e um próximo hop correto para alcançar aquele destino.

remontagem (reassembly)

O procedimento que um receptor utiliza para recriar uma cópia de um datagrama original a partir dos fragmentos que chegam. *Ver fragmentação.*

replay

Uma condição em que a chegada de uma cópia de um pacote antigo confunde a comunicação. Por exemplo, se uma cópia de um pacote que solicita a terminação de uma comunicação é atrasada até depois do início de uma nova comunicação, o pacote pode causar a terminação incorreta da nova comunicação. Os protocolos devem ser projetados para prevenir que o replay cause problemas.

requisição de echo

Uma mensagem usada para teste e depuração. O programa de ping envia mensagens de requisição de echo ICMP para gerar respostas de echo. *Ver resposta de echo.*

resolução de endereço

O mapeamento de um endereço para outro, normalmente de um endereço de alto nível (por exemplo, um endereço IP) para um endereço de baixo nível (por exemplo, um endereço de Ethernet).

resposta de echo (echo reply)

Uma mensagem usada para teste e depuração. Uma resposta de echo ICMP é retornada em resposta a uma mensagem de requisição de echo ICMP. O programa de ping recebe respostas de echo. *Ver requisição de echo.*

retransmissão

A retransmissão de um pacote que foi enviada previamente. Os protocolos de transporte utilizam retransmissão para obter confiabilidade. *Ver PAR.*

retransmissão adaptativa

A habilidade de um protocolo de transporte de mudar continuamente seu temporizador de retransmissão para acomodar variações em atrasos da inter-rede. TCP é o protocolo mais conhecido que utiliza retransmissão adaptativa.

RF (Radio Frequency, ou Freqüência de Rádio)

Uma faixa de freqüências usada para envio de sinais de rádio através do ar (por exemplo, de uma estação de rádio comercial). Tecnologias de rede sem-fio utilizam RF.

RFC (Request for Comments)

Os documentos nos quais são publicados padrões do protocolo TCP/IP.

RIP (Routing Information protocol)

Um protocolo que usa o enfoque vetor-distância para propagar informações de roteamento dentro de um Sistema Autônomo.

RJ-45 (Registered Jack 45)

O tipo de conector usado com Ethernet de par trançado.

RMI (Remote Method Invocation)

A versão orientada a objeto da chamada de procedimentos remotos. *Ver RPC.*

rota default

Uma entrada coringa em uma tabela de roteamento. O software de roteamento segue a rota default se a tabela não contém um rota explícita para o destino.

servidor

Quando dois programas se comunicam através de uma rede, um cliente é aquele que inicia a comunicação, enquanto o programa que espera para ser contactado é um servidor. Um determinado programa pode agir como um servidor para um serviço e um cliente para outro.

servidor de rotas

Um servidor que contém informação de roteamento completa para a Internet global. Todos os servidores de rotas juntos formam o sistema árbitro de roteamento.

servidor raiz

Um servidor de domínio de nomes que conhece as localizações dos domínios de níveis mais altos como *.com* e *.edu*. Ver DNS e domínio.

serviço 2B+D (2B+D service)

Um serviço de ISDN que inclui duas conexões telefônicas padrão mais uma conexão de dados.

signaling

Um termo da companhia telefônica para os mecanismos que fornecem estabelecimento e término da chamada.

síncrono

Característica de qualquer sistema de comunicação em que o remetente deve coordenar (isto é, sincronizar) com o receptor antes de remeter dados. A sincronização é normalmente tratada fazendo-se com que o hardware remetente transmita um pulso regular quando nenhum dado estiver disponível. O receptor usa os pulsos para extrair dados do sinal recebido. Ver assíncrono.

SIP (Session Initiation Protocol)

Um protocolo desenvolvido pela IETF que fornece signaling para sistemas de telefonia IP.

sistema de arbitragem de roteamento

O sistema de servidores de rota onde cada um deles contém informação completa sobre todas as destinações na Internet global.

SMDS (Switched Multi-megabit Data Service)

Uma tecnologia de Rede de Longo Alcance sem conexão oferecida por companhias telefônicas.

SMTP (Simple Mail Transfer Service)

O protocolo usado para transferir e-mail de um computador para outro através da Internet. O SMTP é parte do suíte de protocolos TCP/IP.

SNAP (SubNetwork Attachment Point)

A parte do cabeçalho IEEE LLC/SNAP usada para identificar o tipo de um pacote. O cabeçalho interno é de 8 octetos, com a porção do SNAP ocupando os últimos cinco. Ver LLC.

sniffer

Um sinônimo para analisador de rede, tirado de um produto popular.

SNMP (Simple Network Management Protocol)

O protocolo que especifica como uma estação de gerência de rede se comunica com software de agente em dispositivos remotos como roteadores. O SNMP define o formato das mensagens e seu significado. Ver MIB.

SOAP (Simple Object Access Protocol)

Uma tecnologia middleware orientada a objetos.

SONET (Synchronous Optical NETwork)

Um padrão para codificação digital usado pelas companhias telefônicas.

SPF (Shortest Path First, ou Primeiro Caminho Mais Curto)

Um algoritmo geral de estado de link que os roteadores podem utilizar para computar rotas. Ver estado de link e vetor-distância.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

Constantes e itens numéricos

1 para 1 (1-to-1) [66, 341-342](#)
1 para muitos (1-to-many) [66, 341-342](#)
10/100 Ethernet [573-608](#)
10/100 Ethernet NIC [157-158](#)
1000BaseT [153, 573-608](#)
100BaseT [153, 573-608](#)
10Base2 [152](#)
10Base5 [149-150](#)
10Baset [153, 573-608](#)
125 segundos [186-187](#)
125 μ segundos [179-180](#)
127 endereço [281-282](#)
128 Kbps [187-188](#)
13 Mbps [190-191](#)
1962 [179](#)
2 fios (2-wire) [96-97](#)
200 486-488
2430 bits [186-187](#)
26 Mbps [190-191](#)
2B+D [187-188](#)
32 Kbps [189-190](#)
3-way handshake) [352-354, 573-608](#)
4 fios (4-wire) [93-94](#)
4.1325 KHz [189-190](#)
404 486-488
52 Mbps [190-191](#)
576 Kbps [189](#)
6.144 Mbps [189](#)
6.4 Mbps [189-190](#)
64 Kbps [192-193, 187-188](#)
640 Kbps [189-190](#)
800 [142](#)
[802.11b 124-125, 573-608](#)
[802.16 194-195](#)
[802.1x 557, 573-608](#)
[802.2 142, 573-608](#)
[802.3 573-608](#)
802.3af 462
802.5 [573-608](#)
810 bits [186-187](#)
amostra de 8 bits [179-180](#)
circuito de 3 fios (3-wire circuit) [573-608](#)
circuito de 4 fios (4-wire circuit) [573-608](#)

modelo de referência de 5 camadas [265-267, 573-608](#)
modelo de referência de 7 camadas [244-245, 573-608](#)
serviço 2B+D (2B+D service) [573-608](#)
troca 2-3 (2-3 swap) [83-85, 573-608](#)

A

AAL5 [224-225, 573-608](#)
ABR [223-224, 573-608](#)
Abstract Syntax Notation.1 [540-541, 573-608](#)
Abstract Window Toolkit [514-515, 573-608](#)
accept [402-403](#)
acesso a arquivo [475-476](#)
acesso à Internet [512-513](#)
acesso de arquivo remoto [465](#)
Acesso Direto à Memória [147-149](#)
Acesso Protegido por Wi-Fi [557](#)
ACK [250-251, 573-608](#)
acknowledgement (confirmação) [250-251, 350, 573-608](#)
ACL [549-550, 573-608](#)
adaptador de rede [147-149, 573-608](#)
adaptativo [112-113, 189-190, 351](#)
Adaptive Pulse Code Modulation [180-181](#)
Address Mask Reply [334](#)
Address Mask Request [334](#)
ADSL [188-189, 573-608](#)
Advance Window Toolkit [514-515](#)
aeroporto [124-125](#)
AF_INET [401-402](#)
Agência de Projetos de Pesquisa Avançados [37-38, 209-212, 264-265](#)
agente [539-540](#)
agente de atraso [568, 570-571](#)
AirLAN [124-125](#)
alta velocidade [236](#)
Alternative Window Toolkit [514-515](#)
alto nível [511-512](#)
aluga (lease) [567-568](#)
alvo [294-295](#)
AM [89-90](#)

amarração de endereço [288-289](#)
ambiente em tempo de execução [511-512](#)
American National Standards Institute [160](#)
analizador [142-143](#)
analizador de desempenho [142-143](#)
analizador de pacotes [142-143, 573-608](#)
analizador de rede [59-60, 573-608](#)
âncora [485](#)
anel *ver* IBM Token Ring
anel em forma de estrela [159-160](#)
anel em uma caixa [158-159](#)
anônimo [468-469, 573-608](#)
ANSI [159-160](#)
antena [195-196](#)
antena parabólica receptora [195-196](#)
anúncio de janela [352-353](#)
anycast [325-326](#)
APCM [180-181, 197, 573-608](#)
API [48-49, 397-398, 573-608, 573-608](#)
aplicação eco [52](#)
aplicações [47](#)
apontador de instrução [516](#)
apontar e clicar (point and click) [479](#)
apostas (bidding) [573-608](#)
applet [511-512, 573-608](#)
applet de tipo [517-518](#)
Applet Widget Toolkit [514-515](#)
AppleTalk [246-248, 573-608](#)
Application Program Interface (Interface de Programa Aplicativo) [48-49, 397-398](#)
área [377-378, 573-608](#)
área de dados [138-139](#)
argc [53-54](#)
argumentos [528-529](#)
argv [53-54](#)
armazenamento e encaminhamento [201-202, 573-608](#)
ARP [292, 573-608](#)
ARPA [37-38, 209-212, 264-265](#)
ARPANET [37-38, 209-212](#)
árvore [172-173](#)
árvore distribuída de expansão mínima [172-173, 573-608](#)
AS [370-371, 573-608](#)

- ASCII 80-81, 103-104, 470, 573-608
 ASN.1 458-459, 540-541, 573-608
 ASP 505-506, 573-608
 assinatura digital 550-551, 573-608
 Associação das Indústrias Eletrônicas 80-81, 159-160
 Asymmetric Digital Subscriber Line 188-189
 Asynchronous 79-80, 180-181, 239-240, 573-608
 Asynchronous Transfer Mode 128-129, 213-214, 217-218, 261-262
 ATM 128-129, 213-214, 217-218, 235-236, 261-262, 573-608
 Adaptation Layer 5 224-225
 cabecalho das células 218-219
 segmentação e montagem 224-225
 switch 220-221
 VPI/VCI 220-221
 atraso 236, 252, 371-372
 atraso de acesso 237, 573-608
 atraso de enfileiramento 237, 573-608
 atraso de ida-e-volta 351
 atraso de propagação 236, 573-608
 atraso de switching 234-235, 573-608
 Attachment Unit Interface 150-151
 áudio digital 179-180
 AUI 150-151, 573-608
 cabô 150-151
 conector 150-151
 autenticação 458-459, 548-549
 auto-identificável 138-139, 274-275
 autonegociação 157-158, 573-608
 autoridade de nomeação 428-429
 autoridade para atribuição de nomes 428-429
 autorização 458-459, 548-549
 autorização para enviar (clear to send) 83-84
 autosensing 157-158
 Available Bit Rate 223-224
 AWT 514-515, 573-608
- B**
- backoff 123-124
 backoff exponencial 123-124
 backoff exponencial binário 123-124, 573-608,
 baixa velocidade 236
 barramento em forma estrela 155-156, 159-160
 barramento na caixa 155-156
 Base de Informação de Gerenciamento 541-542, 573-608
 Basic Encoding Rules 458-459
 Basic Rate Interface 187-188
 baud 573-608
 Bearer 573-608
 BER 458-459
 Berkeley Software Distribution 282-283
 BGP 372-373, 573-608
 Biblioteca 511-514
 big endian 535-536
 bin 497-498
 bind 400-401, 573-608
 bit de parada 81-82
 bit de paridade 106-107, 249, 573-608
 bits por segundo 237
 bloqueante 418-419
 bluetooth 124-125, 573-608
 BOOTP 565, 568, 573-608
 Bootstrap 474-475, 561, 573-608
 bps 237, 573-608
- BREAK 82-83
 BRI 187-188, 573-608
 bridge 167, 573-608
 adaptativa 168
 endereço de 172-173
 satélite 170-171
 bridge tap 190-191
 bridges que aprendem (learning bridges) 168
 broadcast 136-137, 195-196, 225-226, 230-231, 573-608
 broadcast de Berkeley 282-283
 Broadcast de Caminho Reverso 379-380
 broadcast direcionado 573-608
 BSD UNIX 282-283, 466-467
 bufferização 171-172
 bytecode 512
- C**
- C++ 512
 cabeamento 158-159
 cabeamento Cat5 157-158
 cabecalho 133-134, 138-139, 248-249, 438-439, 573-608
 base 321-322
 de extensão 321-322, 573-608
 opções 323-324
 cabecalho de início 103-104
 cable modem (modem de cabo) 573-608
 cable Modem Termination System 194-195
 cabo ver fio
 cabo coaxial 71-72, 573-608
 cabo de categoria 5 573-608
 cache 295-296, 489
 Cache ARP 295-296
 caching 431-432, 489-490
 caixa 158-159
 Cálculo de rota distribuída 200
 call forking (chamadas para múltiplos locais) 460-461
 camada 244-245
 de transporte 339-340
 camada de interface de rede 297
 Camada de Soquete Segura 557
 caminho duplo 184
 caminho mais curto 207-208
 campo 133-134
 campo de tipo 295-296
 canal 73-74, 189-190
 canal B 187-188, 573-608
 canal compartilhado 117-118
 canal D 187-188, 573-608
 canal virtual 219-220, 573-608
 Canal Virtual Comutado (Switched Virtual Channel, SVC) 173-174, 197, 233, 573-608
 canal virtual permanente 222
 cancelamento do eco 221
 capa 615
 Capacidade 252
 caracteres 80-81
 característica 458-459
 Carrier Sense Multiple Access 122-123, 573-608
 cartão de interface de rede 147-149, 573-608
 CAT 363-364, 573-608
 categorias de cabeamento 157-158
 CATV 191-192, 573-608
 CBR 223-224, 573-608
 CBT 380-381, 573-608
 CCITT 212-213, 573-608
 CD 123-124
- CD-ROM 613
 cell tax 219-220, 225-226, 573-608
 célula 218-219, 573-608
 CGI 497-498, 506, 573-608
 CGI bin 497-498
 chamada de procedimento 527-528
 chamada do método 532-533
 Chamada Remota de Procedimento 527-528
 chamadas de trabalhos 453-454
 Channel Service Unit 181-182
 character stuffing (enxerto de caracteres) 104-105
 CHARGEN 420-421
 chat 56-57, 573-608
 chave 549-551
 chave de cifragem 549-550, 573-608
 chave privada 550-551
 chave pública 550-551
 chave secreta 548-549
 checksum 107-108, 548-549, 573-608
 Chicago 179
 ciclos por segundo 85
 CIDR 573-608
 cifragem 232, 549-550
 circuito alimentador 193-194, 573-608
 circuito de dados seriais 93
 circuito de três fios 83-84
 circuito de tronco 573-608
 circuito virtual 220-221
 ver também canal virtual
 circuito virtual comutado 222-223
 circuito virtual permanente 234-235, 573-608
 CL 232, 573-608
 classe 273-275, 511-512
 classe de endereço 273-275
 classe de tráfego 321-322
 classe primária 273-275
 cliente 47-48, 386-387, 573-608
 Cliente Agente do Usuário 456-457
 cliente de eco em echoclient.c 54-55
 cliente do chat em chatclient.c 59
 cliente.c 413-414
 clocked 180-181
 close 398-401, 406-407
 closesocket 400-401, 420-421
 cluster 325-326, 573-608
 CMTS 194-195, 573-608
 CNAME 432-433
 CO 187-188, 232, 573-608
 coaxial 71-72
 Coaxial de Fibra Híbrido 193-194
 codebase 518-519
 Codificação Manchester 120-121, 573-608
 codificar 440-441
 código 518-519
 código de autenticação de mensagem 548-549
 colaboração 319-320
 colapso por congestionamento 255-256, 354-355
 ColdFusion 505-506, 573-608
 coleta de lixo automática 512-513
 colisão 122-123, 573-608
 collapsed backbone 155-156
 collision avoidance 124-125
 collision detection 123-124, 573-608
 colon hex 326-327
 coluna (SONET) 186-187
 COM 534
 Common Gateway Interface 497-498, 506, 573-608

Common Object Request Broker Architecture 533
 community Antenna TeleVision 191-192
 compilador 510-511
 compilador JAVA 516
 component Object Model 534
 compressão de zeros 326-327, 573-608
 computador da classe servidor 387-388
 comutação de circuito 233, 573-608
 comutação de rótulo 220-221, 573-608
 Comutação de Rótulo Multiprotocolo 226
 comutação IP 573-608
 comutação na camada 3 573-608
 concatenado 185-186
 concorrente 390
 conditional *get* 491
 conector
 DB-15 83-84
 DB-25 83-84
 DB-9 83-84
 RJ-45 153-154
 conector BNC 152, 573-608
 conector DB 573-608
 conexão 219-220
 conexão de controle 467-468
 conexão de dados 473-474
 conexões persistentes 489-490
 conexões virtuais 348-349
 confiável 250-251
 confidencialidade 232, 547-550
 confidencialidade de dados 547-548
 configuração de protocolo 561-562, 573-608
 configurado 561-562
 confirmação positiva com retransmissão 250-251
 congestionamento 237-238, 255-256, 573-608
 connect 402-403
 Constant Bit Rate 223-224
 contagem de hops 372-373, 573-608
 contagem de referências 406-407
 Controlador de Gateway de Mídia 454-456
 controle de congestionamento 354-355
 controle de fluxo 251-252, 573-608
 controle de taxa 255-256
 conversor A-D 179-180
 conversor analógico-digital 89-90
 conversor D-A 179-180
 conversor digital-analógico 179-180
 convidado 468-469
 cookie 500-501, 573-608
 cópia cega em carbono 440-441
 copper 71
 CORBA 5 110-111, 573-608
 corpo 438-439, 480-481
 corrente 79-80
 CPU 147
 CRC 108-111, 249, 573-608
 criação de objeto dinâmico 512
 CR-LF 443-444
 CRLF 62-63, 485-486
 CSMA 122-125, 573-608
 CSMA/CA 124-125
 CSMA/CD 123-124, 573-608
 CSU 181-182
 CTS 83-84

D

DARPA 37-38
 Data Over Cable System Interface Standard 194-195
 Data Service Unit 181-182

data stuffing (enxerto de dados) 104-105
 datagrama 573-608
 datagrama do usuário 343-344
 datagrama IP 302-303, 573-608
 DAYTIME 419-420
 dB 85-86
 DB-15 83-84
 DB-25 83-84
 DB-9 83-84, 87
 DCE 83-85, 533, 573-608
 DCE/RPC 533
 DCOM 534
 decibéis 85-86
 demarc 182-183, 573-608
 demodulador 92-93, 573-608, 573-608
 demultiplexar 573-608
 descoberta de roteador 563-564
 descriptor 398-399
 desempenho efetivo 237
 desordenado 249
 DESTINATION ADDRESS 322
 DESTINATION PORT 391-392
 destination unreachable 332-333
 destino 280-281, 552-553
 DHCP 566-567, 573-608
 diagrama de formato de onda 80-81
 dicionário 513-514
 difusão limitada 281
 Digital Equipment Corporation 533
 Digital Subscriber Line 188-189
 digitalização 179-180
 diodo emissor de luz (light emitting diode) 72-73
 disparado pelo limite (edge triggered) 120-121
 disponibilidade 547-548
 disponibilidade de dados 547-548
 distância 209
 Distributed Component Object Model 534
 Distributed Computing Environment 533
 divisão em camadas 297
 DIX Ethernet 120-121, 573-608
 DMA 147-149
 DMT 189-190
 DNS 423-424, 573-608
 DNS reply 431
 DNS request 431
 DOCSIS 194-195, 573-608
 documento ativo 495, 510, 573-608
 documento da Web 495
 documento dinâmico 495-496, 573-608
 documento estático 495, 573-608
 DOM 522
 Domain Name System (Sistema de Nome de Domínio) 423-424, 573-608
 domínio 573-608
 Domínio Administrativo de Telefone IP 460-462
 domínio de alto nível 423-424
 domínio registrar 424-425
 DOS 573-608
 downlink 573-608
 download 465-466
 downstream 188-189, 573-608
 DS-0, DS-1, DS-3 573-608
 DSO 184
 DSL 188-189, 573-608
 DST 172-173, 573-608
 DSU/CSU 181-182, 573-608
 DTE 83-85, 573-608
 DTMF 458, 573-608
 Dual Tone Multi-Frequency (DTMF) 458
 duplicação 250-251
 duração da conexão 233
 DV 573-608
 DVMRP 380-381, 573-608
 DWDM 96-97, 573-608
 Dynamic Host Configuration Protocol 566-567

E

E/S de rede 513-514
 E1 183-184
 E2 183-184
 E3 183-184
 EBCDIC 470, 573-608
 ECMA script 520-521
 eco 342-344
 EGP 371-372, 573-608
 EIA 80-81, 159-160
 e-mail 437, 573-608
 cabecalho de 438-439
 corpo de 438-439
 endereço de 437-438
 gateway 445-446
 interface de 442
 relay 445-446
 emendas TCP (TCP splicing) 362-363, 573-608
 encaminhador de mensagens 444-445
 encaminhamento 305
 encapsulamento 294-295, 311-312, 573-608
 endereçamento com classes 273-275, 573-608
 endereçamento de subrede 189-190
 endereçamento hierárquico 202-203, 573-608
 endereçamento sem classes 277-278, 573-608
 endereço 195-196, 573-608
 com tudo em um, 281
 com tudo em zero 281-282, 573-608
 de destino 133-134, 168, 573-608
 de fonte 168
 de loopback (127) 281-282
 de prefixo de rede 281
 de protocolo 287-288, 573-608
 endereço de acesso a mídia 133-134
 endereço de broadcast 137-138, 140-141, 573-608
 endereço de broadcast direcionado 281
 endereço de hardware 133-134, 287, 573-608
 endereço de Internet 280-281, 573-608
 endereço de Protocolo Internet 280-281
 endereço deste computador 281-282
 endereço endpoint 402-403, 573-608
 endereço físico 133-134, 168, 287, 573-608
 endereço fonte 133-134, 573-608
 endereço IP 280-281, 573-608
 endereço MAC 133-134, 573-608
 endereço unicast 137-138
 end-of-line 443-444
 endpoint 339-340
 ENUM 460-462
 envelope 534
 enxerto de bits (bit stuffing) 104-105, 125-126, 182-183
 enxerto de bytes (byte stuffing) 104-105, 573-608
 eot 103-104
 Equipamento de Comunicação de Dados 83-85
 Equipamento Terminal de Dados 83-85

erro 111
 erro de enquadramento 82-83, 573-608
 erro de rajada 111
 erro de transmissão 105-106, 573-608
 erro vertical 111
 escalabilidade 199-200
 Escritório Central 187-188
 espectro espalhado 96-97, 573-608
 espera de chamadas 453-454
 esqueleto 505-506
 esquema de empacotamento (COM) 534
 esquema de endereçamento configurável 135-136
 esquema de endereçamento dinâmico 135-136
 esquema de endereçamento estático 135-136
 esquema de endereço 233-234
 estabilidade 378
 estação 133-134
 estrela 128-129
 ether 120-121
 Ethernet 120-121, 234-235, 573-608
 10Base5 149-150
 AUI 150-151
 conector BNC 152
 de fio espesso 149-150, 573-608
 de fio fino 152, 573-608
 endereço 144-145
 endereço broadcast 140-141
 endereço multicast 140-141
 hub 153
 quadro 139-141
 repetidor 164-165
 segmento 165-166
 tipos 140-141
 topologia 155-156
 Ethernet de par trançado 153, 573-608
 Etherreal 144-145
 etiqueta applet 518-519
 etiqueta IMG 483-484
 etiquetas 480-481, 534-535
 exceção 513-514, 520-521
 execução concorrente 391
 execução interpretativa 512
 execução multithreaded 513
 extremidade 205-206

F

fábrica 515-516
 família de protocolo 244
 Fast Ethernet 120-121, 573-608
 favorito 491-492
 FDDI 127-128, 573-608
 FDM 94-95, 573-608
 ferramenta 526-527
 ferramenta de software 526-527
 ferramentas gráficas 514-515
 fetch 477
 Fiber Distributed Data Interconnect 127-128
 Fiber Optic Intra-Repeater Link 165-166
 Fiber To The Curb 194-195
 fibra 127-128, 573-608
 fibra ótica 72-73, 573-608
 File Transfer Protocol 427-428, 465-466
 filtro de pacote 552-553, 573-608
 fim da transmissão 103-104
 fim de arquivo (end-of-file) 48-49
 fim-a-fim 339-340, 348-349, 491-492, 573-608
 FIN 573-608
 fio 71
 coaxial 71-72
 firewall 551

firewall Internet 551, 573-608
 FLOW LABEL 322-323
 fluxo 573-608
 FM 89-90
 FOIRL 165-166
 fonte 280-281, 552-553
 forma 118
 formulário 504-505
 fractional T1 184, 573-608
 fragmentação 314, 324, 573-608
 fragmento 573-608
 Frame Relay 212-213, 234-235, 573-608
 freqüência 73-74
 FTP 427-428, 465-466, 573-608
 comandos 466-467
 FTTC 194-195, 573-608
 full duplex 83-84, 92-93, 233, 573-608
 função universal 407-408

G

garantias de serviço 217-218, 233-234
 Gatekeeper 455-456, 457
 Gateway de Mídia 455-456
 gateway de sinalização 455-456
 Gbps 120-121, 237, 573-608
 GEO 74-75, 573-608
 gerência de rede 573-608
 gerente 539-540
 gerente de lista 446
 gerente de rede 539
 GET (HTTP) 62-63, 485-486
 gethostbyaddr 405-406
 gethostbyname 405-406, 431
 gethostname 405-406
 getpeername 405-406, 421
 getsockop 405-406
 GIF 483-484, 573-608
 Gigabit Ethernet 120-121, 573-608
 gráficos 512-514
 grãos finos 223-224
 graphics interchange format 483-484
 ver também GIF

H

H.323 454-455, 457-458, 573-608
 half duplex 83-84, 93-94, 573-608
 hardware de interface 147
 hardware de rede 147
 hashing 548-549
 hashing criptográfico 548-549
 HDSL 190-191, 573-608
 HDSL2 190-191
 head 480-481
 HEAD 485-486
 headroom 157-158
 Hertz 85, 573-608
 HFC 193-194, 573-608
 hidden station problem 124, 598
 High-Rate Digital Subscriber Line 190-191
 hipermídia 479-480, 573-608
 hipertexto 479-480, 573-608
 homepage 479-480, 573-608
 hop 43-44, 203-204, 374-376, 573-608
 hospedeiro 267-268, 573-608
 HTML 479-480, 573-608
 HTTP 485-486, 573-608
 HTTP-NG 489-490
 HTTPS 489-490, 573-608
 Hub 119, 159-160, 573-608
 HyperText Makup Language 479-480
 HyperText Transfer Protocol 485-486
 Hz 85

I

I/O de rede 513-514
 IANA 573-608
 IBM Token Ring 126-127, 573-608
 ICANN 424-425, 573-608
 ICMP 331-332, 563-564, 573-608
 tipo 333
 ICS 363-364, 573-608
 ID da chamada 453-454
 identificador da conexão 235-236
 IDL 532-533, 573-608
 IDS 555-557, 573-608
 IEEE 80-81, 142
 802.2 142
 autoridade para registro 144-145
 cabeçalho LLC/SNAP 142
 IETF 453-454, 573-608
 IGMP 378-379, 573-608
 IGP 371-372, 573-608
 IIS 505-506, 573-608
 IMP 209-212
 importa 517
 INADDR_ANY 402
 inalcançável 332-333
 independência de fonte 204
 independente de protocolo 380-381
 informações de estado 500-501
 infravermelho 76
 Institute for Electrical and Electronic Engineers 80-81
 Integrated Services Digital Network 187-188
 integridade 547-548
 integridade de dados 547-548
 interface de mensagem 233-234
 interface de programa 573-608
 interface de stream 233-234
 Interface Definition Language 532-533
 Interface Message Processor 209-212
 interferência eletromagnética 105-106
 Interferência Inter-símbolo 85
 International Organization for Standardization 244-245
 International Telecommunications Union 80-81, 453-454
 Internet 37-38, 264-265, 573-608
 Internet Assigned Number Authority 277-278
 Internet Connection Sharing 363-364
 Internet Control Message Protocol 331-332
 Internet Engineering Task Force 453-454
 Internet global 264-265
 Internet Group Multicast Protocol 378-379
 Internet Information Server 505-506
 Internet pública 264-265
 Internet Service Provider 277-278, 573-608
 Internet Softswitch Consortium 457
 interpretador 512
 inter-rede 262-263
 intranet 573-608
 IP 280-281, 573-608
 IP-in-IP 379-380, 555-556, 573-608
 ipInReceives 542-543
 IPng 320-321, 573-608
 ipRouteNextHop 543-544
 IPsec 557, 573-608
 IPv4 320-321, 573-608
 IPv6 320-321, 573-608
 IPX 140-141
 ISC 457, 573-608
 ISDN 187-188, 197, 217-218, 573-608
 ISO 244-245, 573-608

isochronous 180-181, 239-240, [573-608](#)
 ISP 277-278, [573-608](#)
 ITAD 460-462
 ITU [80-81](#), [209-212](#), [453-454](#), [573-608](#)

J

janela 350-353, [573-608](#)
 janela de tamanho zero 352-353
 janela deslizante 252, [573-608](#)
 Java 511-512, [573-608](#)
 Java RMI 534, [573-608](#)
 Java Server Pages 505-506
 javac 516
 JavaScript 520-521, [573-608](#)
 Jini 534
 jitter 217-218, 238-239, [371-372](#), [573-608](#)
 JPEG [573-608](#)
 JSP 505-506, [573-608](#)

K

Kbps [124-125](#), [573-608](#)

L

LAN [117-118](#), 199, [573-608](#)
 LAN sem-fio [124-125](#), [573-608](#)
 largura de banda [85](#), 237, [573-608](#)
 laser [72-73](#), [76-77](#)
 LED [72-73](#)
 Lei de Kepler [74-75](#)
 Lei de Shannon-Hartley [85](#)
 LEO [75-76](#), [573-608](#)
 ligação inter-rede [37-38](#), [262-263](#)
 limite de descida [120-121](#)
 limite de hops 322-323, [573-608](#)
 limite de subida [120-121](#)
 linefeed 443-444, [467-468](#), 485-486
 Linguagem de Programação Java 511-512
 linguagem markup 479-480
 Linguagem Markup Extensível 489-490
 linha de assinante local [187-188](#)
 linha serial [93](#), [573-608](#)
 linha serial alugada [93](#)
 link 205-206
 lista de controle de acesso 549-550
 lista de mensagens 444-445
 lista não-ordenada [482-483](#)
 listen 402
 little endian 535-536
 LLC [142](#), [573-608](#)
 Local Area Network [117-118](#), 199, [573-608](#)
 localhost 411-412
 localidade de referência [117-118](#), [429-430](#), [489](#), [573-608](#)
 localidade espacial [118](#)
 localidade física [118](#)
 localidade temporal [118](#)
 LocalTalk [124-125](#), [158-159](#)
 Logical Link Control [142](#), [573-608](#)
 long-haul network 199, [573-608](#) ver também WAN
 loop de assinante [187-188](#)
 Loop local [187-188](#), [573-608](#)
 loopback 182-183

M

MAC 548-549
 mail exploder 444-445, [573-608](#)
 mailer 450
 MAN [194-195](#), 199, [573-608](#)
 marcador de página 491-492
 MARK [81-82](#)

marshaling 530-531
 máscara 278-279, [573-608](#)
 máscara de endereço 278-279, 303-304, 563-564, [573-608](#), 611
 máscara de subrede 278-279, 563-564, [573-608](#), 611
 Masquerade 363-364, [573-608](#)
 MBONE 380-381, [573-608](#)
 Mbps [120-121](#), 237, [573-608](#)
 MCU 457, 458-459, [573-608](#)
 Megaco 454-457
 melhor esforço (best-effort) 305-306, [573-608](#)
 Memória Somente Leitura 474-475
 mensagem eletrônica 437, [573-608](#)
 endereço de 437-438
 método 512, 532-533, 534
 métrica de roteamento [371-372](#)
 MGCP 454-457, [573-608](#)
 MIB 541-542, [573-608](#)
 microkernel 512-513
 microsegundo [179-180](#)
 microsoft 533
 middleware 532-533, [573-608](#)
 mídia [71](#)
 mídia de transmissão [71](#)
 MIME 441, [573-608](#)
 Modelo de camadas TCP/IP 265-267
 Modelo de Objetos do Documento 522
 modelo de referência Internet 265-267, [573-608](#)
 modelo em camadas 244-245, [573-608](#)
 modelo em camadas Internet 265-267
 modem [83-85](#), [92-93](#), [573-608](#)
 de 2-fios [93-94](#)
 de 4-fios [92-94](#)
 de discagem [93-94](#)
 de fibra [163-164](#), [573-608](#)
 de linha alugada [93](#)
 óptico [93](#), [573-608](#)
 RF 93
 modem dialup [93](#), [573-608](#)
 modem head-end [194-195](#), [573-608](#)
 modo de chamada [93-94](#)
 modo de resposta [93-94](#)
 modo promíscuo [142-143](#), [573-608](#)
 modo silencioso 472-473
 modulação 89-90, [92-93](#), [189-190](#), [573-608](#)
 amplitude de 89-90
 deslocamento de fase de 90-91
 freqüência de 89-90
 modulação da amplitude, 90
 modulação de freqüência 89-90
 modulação Discrete Multi Tone [189-190](#)
 modulação por deslocamento de fase 90-91, [573-608](#)
 modulador [92-93](#), [573-608](#)
 monitor [142-143](#)
 monitor de rede [142-143](#), [573-608](#)
 MOSPF 380-381, [573-608](#)
 MPEG4 [573-608](#)
 MPLS [226](#), [573-608](#)
 mrouted 380-381, [573-608](#)
 MSRPC 533
 MSRPC2 533
 MTU 312-314, 325, [573-608](#)
 MTU de caminho 317, 325, 336-337, [573-608](#)
 muitos-para-1 341-342
 multicast 138, 325-326, [573-608](#)
 endereço 138, [140-141](#)
 endereço de ponte [172-173](#)
 roteamento 378-379

Multicast backBONE 380-381
 multi-homed [283-284](#), 373-374, [573-608](#)
 multiplexação [573-608](#)
 por divisão de cor 96-97
 por divisão de frequência [94-95](#), 189-190, [573-608](#)
 por divisão de onda ver WDM
 por divisão de tempo 97-98, [102-103](#), [184-187-188](#), [576-608](#)
 multiplexação estatística 97-98, [573-608](#)
 multiplexação inversa [189-190](#)
 multiplexação por divisão de comprimento de onda densa 96-97
 multiplexação por divisão de tempo 97-98, [102-103](#), [184-187-188](#), [573-608](#)
 Multiplexação por Divisão de Tempo em intervalos (Slotted Time Division Multiplexing) 97-98
 Multiplexação por Divisão de Tempo Síncrona 97-98
 multiplexador
 add/drop 186-187
 inverso [184](#)
 multiplexador da conexão [150-152](#)
 Multipurpose Internet Mail Extensions 441
 mux Add/drop 186-187

N

na mesma linha 360
 NAPT 362-363, [573-608](#)
 NAT 360, [573-608](#)
 National Center for Supercomputer Applications 497-498
 National Science Foundation 264-265
 navegador 479, [573-608](#)
 NBMA [225-226](#), [573-608](#)
 NCSA 497-498
 Network File System 475-476, [573-608](#)
 NEXT READER 322-323
 next-hop [209](#), 287-288, [562-563](#)
 next-hop forwarding [203-204](#), [573-608](#)
 NFS 475-476, [573-608](#)
 NIC [147-149](#), [573-608](#)
 conector [149-150](#)
 NIU 182-183, [573-608](#)
 nodo 205-206, [573-608](#)
 nodo de rede 205-206
 nomeação 423
 Non Broadcast Multiple Access [225-226](#)
 nonroutable 360
 notação CIDR 279-280, 611
 notação com barras [573-608](#), 611
 notação de decimal pontilhada [276](#), 573-608
 notação de dois pontos hexadecimal 326-327, [573-608](#)
 nslookup 435
 NTI [573-608](#)
 núcleo 380-381
 null modem [83-85](#)
 número de porta de protocolo 342-344, 390, [573-608](#)
 número de rede 272-273
 Nyquist [85](#)

O

Object RPC 533
 objeto 532-533
 OC 185-186, [573-608](#)
 OC-3 [573-608](#)
 ONC OPC 532-533
 ONU [190-191](#)
 opções (IP) 307-308



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

TCP/IP 264-265, [573-608](#)
 tcpdump [144-145](#)
 TDM 97-98, [184](#), 455-456, [573-608](#)
 tecnologia banda base 96-97, [573-608](#)
 tecnologia de broadcast 96-97, [573-608](#)
 telecommunications Industry Association
[159-160](#)
 telecommute 231-232, 554-555
 telefone IP 454-456
 telefonia 453
 telefonia IP 453, [573-608](#)
 televisão a cabo [191-192](#)
 telnet 419-420
 template 505-506
 tenex [467-468](#)
 teorema de amostragem [179-180](#)
 Teorema de Amostragem de Nyquist 573-
 608
 Teorema de Shannon [85](#)
 terminador [150-152](#), [573-608](#)
 terminador de cabo [150-152](#)
 Terminal 457
 terra [83-84](#)
 TFTP [474-475](#), 566-567, [573-608](#)
 Thicknet [149-150](#), [573-608](#)
 Thinnet 152, [573-608](#)
 throughput 237, 252, [371-372](#), [573-608](#)
 TIA [159-160](#)
 Time Division Multiplexing 455-456
 Time Exceeded [332-333](#)
 timeserver 342-344
 tipo 139-140, 333, 432-433
 tipo de mensagem (DHCP) [569](#)
 tipo de quadro explícito 138-139
 tipo em quadros [140-141](#)
 TLD 423-424, [573-608](#)
 Token Ring *ver* IBM Token Ring
 topologia [118](#), [147](#), [573-608](#)
 de barramento 119-122, [573-608](#)
 em anel 119, 125-127, 186-187, [573-
 608](#)
 em estrela [118](#), [128-129](#), [573-608](#)
 hub 119
 malha [115-116](#)
 ponto a ponto [115-116](#)
 TP Ethernet [153](#), [573-608](#)
 traceroute [42-44](#), 334-335, [573-608](#)
 tracert [43-44](#), 335-336, [573-608](#)
 Tradução de Endereço de Cabo 363-364
 Tradução de Endereço de Rede 360
 Tradução de Porta e Endereço de Rede 497-
 498
 trailer [248-249](#)
 transceiver [150-151](#), [159-160](#), [573-608](#)

transferência de arquivo [465](#) *ver* também FTP
 transferência de mensagens 442
 trânsito 373-374
 transmissão paralela [80-81](#)
 transmissão simplex [83-84](#), 233
 Transmission Control Protocol 339-340,
 347-348, [573-608](#)
 transponder [73-74](#)
 tratamento de erros 520-521
 TRIP 460-462
 Trivial File Transfer Protocol 474-475
 troca de mensagem [291-292](#)
 tronco 185, 193-194
 TTL [573-608](#)
 túnel [379-380](#), 555-556, [573-608](#)
 túnel IP 555-556, [573-608](#)
 TV interativa 193-194
 Twice NAT 363-364

U

UDDI 534-535, [573-608](#)
 UDP 335-336, 339-340, 391-392, 532-533,
[573-608](#)
 UDP CHECKSUM 343-344
 UDP DESTINATION PORT 343-344
 UDP MESSAGE LENGTH 343-344
 UDP SOURCE PORT 343-344
 unicast 325-326, [573-608](#)
 Unidade de Controle Multiponto 457, 458-
 459
 unidade de interface de rede 182-183
 unidade de transmissão máxima 312-314,
[573-608](#)
 unidos [187-188](#)
 Uniform Resource Locator 483-484, [573-
 608](#)
 UNIX 413-414, [466-467](#)
 uplink [194-195](#), [573-608](#)
 upstream [188-189](#), [573-608](#)
 URI 459-462
 URL 483-484, [573-608](#)
 utilização 237-238
 UTP [71-72](#), [573-608](#)

V

V.35 182-183
 variáveis de ambiente 499-500
 varredura de portas 555-557
 VBR 223-224, [573-608](#)
 VC [219-220](#), [573-608](#)
 VCI [220-221](#)
 VDSL [190-191](#)
 velocidade [147](#), 237

velocidade do cabo 360-361
 verificação de redundância [108-109](#)
 verificação de redundância cíclica [108-109](#),
 249, 548-549, [573-608](#)
 verificação de tipo estático 512
 VERS [321-322](#)
 Very-high bit rate Digital Subscriber Line
[190-191](#)
 vetor-distância [209](#), [374-375](#), [573-608](#)
 vídeo sob demanda 193-194
 Virtual Channel Identifier [220-221](#)
 Virtual Path Identifier [220-221](#)
 Virtual Private Network 231-232, 554-555
 Voice Extensible Markup Language 489-
 490
 voicemail 453-454
 VoiceXML 489-490
 VoIP 453, [573-608](#)
 voltagem [80-81](#)
 Voz sobre IP 453
 VPI [220-221](#)
 VPI/VCI [220-221](#), [573-608](#)
 VPN 231-232, 554-555, [573-608](#)
 VXML 489-490, [573-608](#)

W

WAN 199, [573-608](#)
 WAP [573-608](#)
 WaveLAN 124-125
 WDM [81-82](#), [573-608](#)
 Web [573-608](#)
 webclient.c [60-61](#)
 webserver.c [62-63](#)
 WEP 557, [573-608](#)
 Wi-Fi [124-125](#), [557](#), [573-608](#)
 wildcard 469-470
 WiMAX [194-195](#)
 Windows NT 413-414
 Windows sockets 400-401
 Wired Equivalent Privacy 557
 World Wide Web 479, [573-608](#)
 WPA 557, [573-608](#)
 Write 398-400, 405-406
 WWW [479](#), [573-608](#)

X

X Window System [515-516](#)
 X.25 [140-141](#)
 xanim 613-614
 XDR 532-533, 535-536, [573-608](#)
 XML 489-490, 534-535, [573-608](#)
 xor [108-109](#)
 xSDL [188-189](#)



Bookman Companhia Editora
Av. Jerônimo de Ornelas, 670
90040-340 Porto Alegre, RS, Brasil
Fone (51) 3027-7000 Fax (51) 3027-7070
e-mail: bookman@artmed.com.br

TECNOLOGIA DA INFORMAÇÃO
Redes

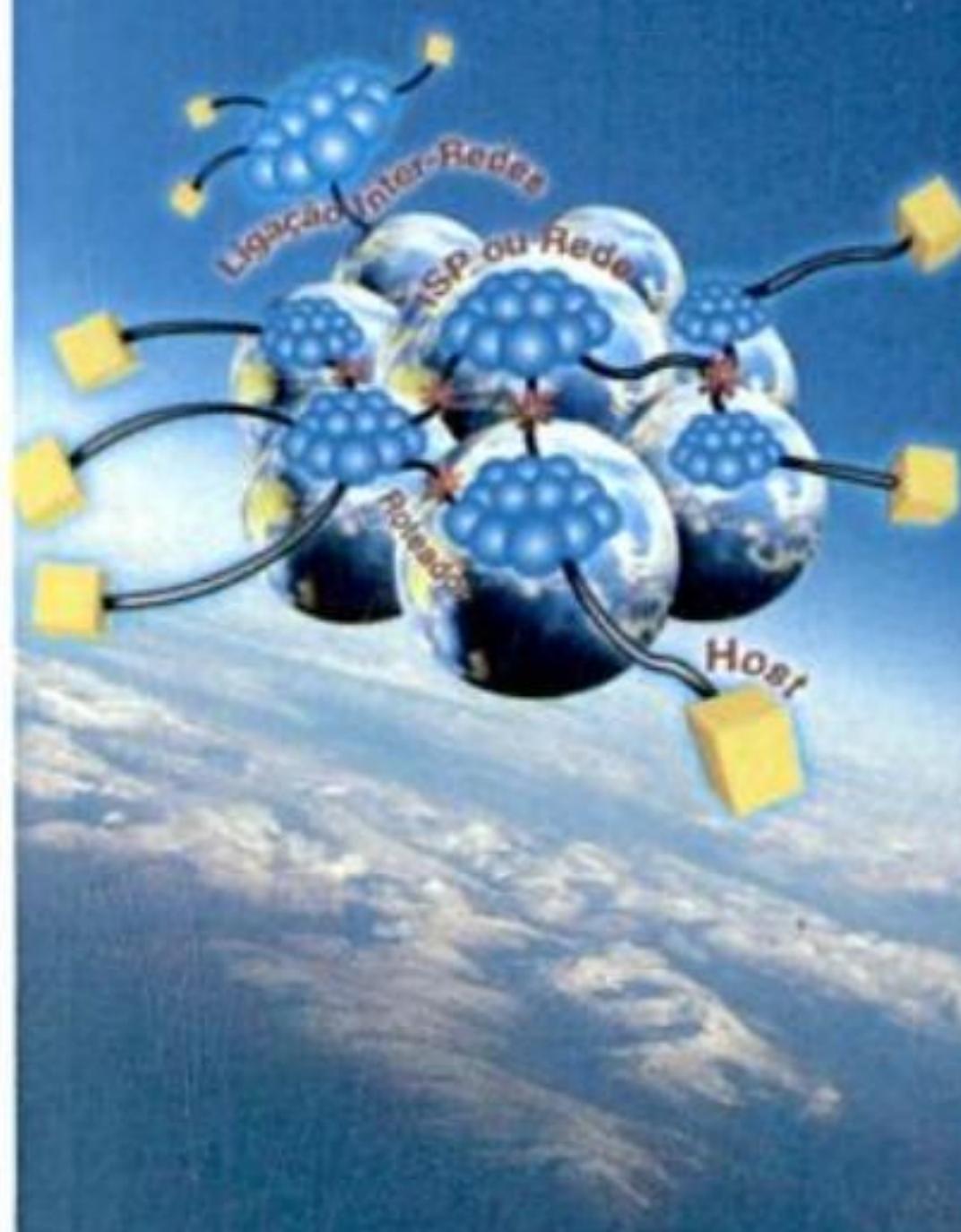
Cheswick, Bellovin & Rubin
Firewalls e Segurança na Internet:
Repelindo o Hacker Ardiloso, 2.ed.

Comer, D.
Redes de Computadores e
Internet: Abrange Transmissão de
Dados, Ligações Inter-redes, Web
e Aplicações
4.ed.

Forouzan, B.
Comunicação de Dados e Redes
de Computadores, 3.ed.

Stevens, Fenner & Rudoff
Programação de Rede UNIX,
Volume 1, 3.ed.

Tittel, E.
Rede de Computadores (COLEÇÃO
SCHAUM)

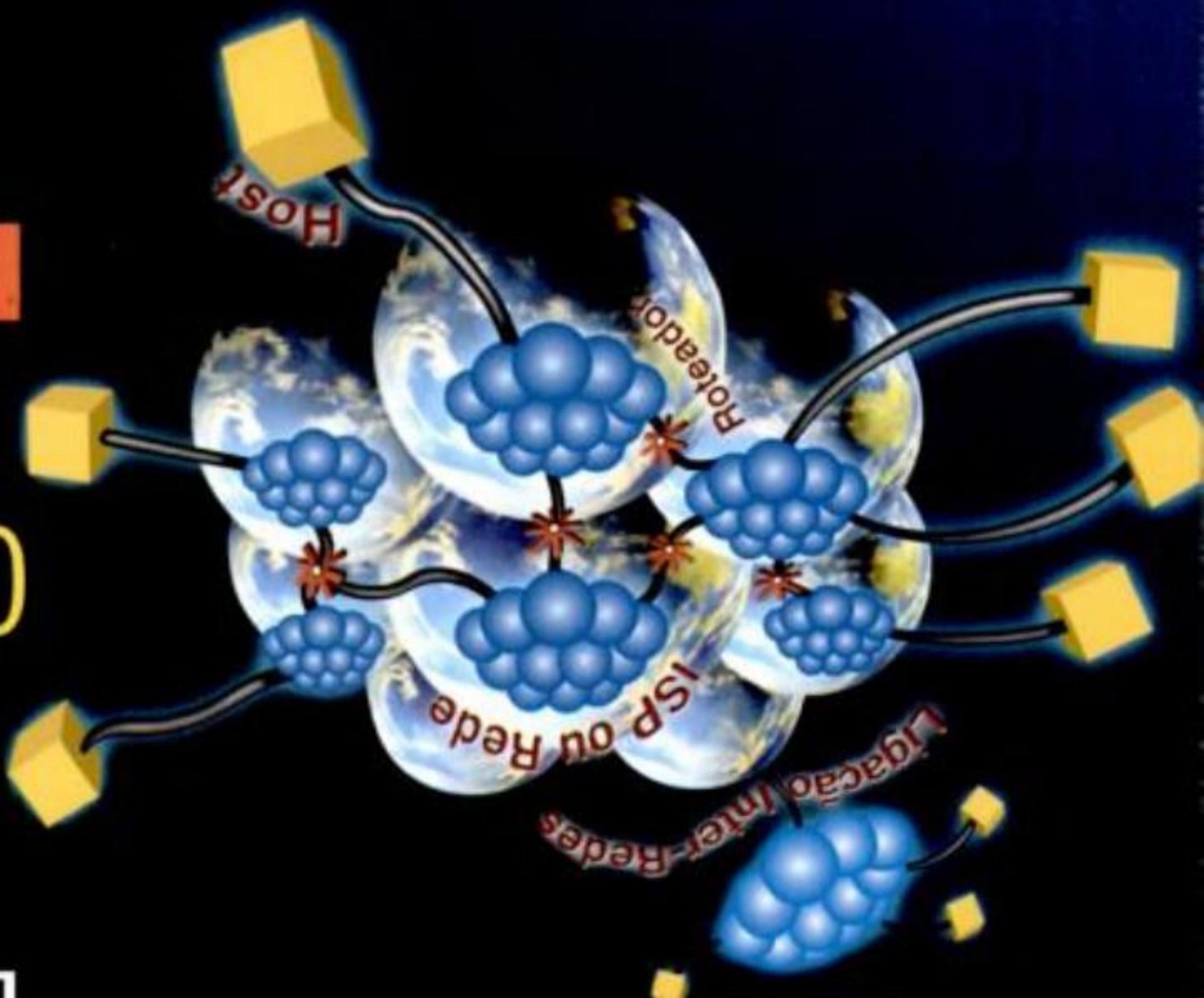




9 78560 031368

8-95-15009-58-8/6 NBST

Douglas E. Comer REDES DE COMPUTADORES COMPUTADORES E INTERNET



«Nao pude parar de ler ate termina-lo. Simplesmente soberbo.»

JOHN LIN, BELL LABS

Autor renomado e uma autoridade em redes de computadores, Douglas Gomer constrói um panorama abrangente de tecnologias que permitem que a Internet forme serviços de aplicação, navegar e troca instantânea de mensagens. Esta edição traz capítulos novos sobre a tecnologia da Internet. O livro oferece um tour incomparável que abrange desde as aplicações da Internet até os baixos níveis da transmissão de pacotes. Mostra como os protocolos são dispostos em camadas e como essas camadas se relacionam.

NOVIDADES DESTA EDIÇÃO:

- Capítulo 24, UDP: Serviço de Transporte de Datagrama: introduz o protocolo de transporte tim-a-tim e mostra como usa-lo. Embora considerado insignificante por alguns, o UDP forma a base para aplicações multicast e broadcast e novas aplicações em áudio e vídeo.
- Capítulo 26, NAT: Tradução de Endereço de Redes: explica como a tecnologia NAT supera a maior limitação da Internet permitindo o compartilhamento de um único endereço IP por vários computadores, especialmente importante para instalações residenciais e para pequenos negócios.
- Capítulo 33, Telefonia IP: Discute a mais recente aplicação da tecnologia da Internet: a telefonia de voz (VoIP). O capítulo expllica padroes concorrentes de telefonia IP, incluindo protocolos como H.323, Session Initiation Protocol (SIP) e Megaco/H.248, e mostra também amostra de sessão SIP.