

COMUNICAÇÃO DE DADOS E REDES DE COMPUTADORES

BEHROUZ A. FOROUZAN

3^a EDIÇÃO



COMUNICAÇÃO DE DADOS E REDES DE COMPUTADORES



F727r

Forouzan, Behrouz A.

Comunicação de dados e redes de computadores / Behrouz A. Forouzan ; tradução Glayson Eduardo de Figueiredo. – 3. ed. – Porto Alegre : Bookman, 2006.

840 p. ; 25 cm.

ISBN 978-85-363-0614-8

1. Sistemas de transmissão de dados. 2. Redes de computadores.
1. Título.

CDU 621.39:004.7

Catalogação na publicação: Júlia Angst Coelho – CRB Provisório 05/05

BEHROUZA FOROUZAN
DeAnza College

COMUNICAÇÃO DE DADOS E REDES DE COMPUTADORES

3^a EDIÇÃO

Tradução:

Glaysom Eduardo de Figueiredo
Mestrando em Engenharia Elétrica – UFMG
Físico pela UFMG
Professor de Comunicação de Dados e Conectividade do CETEL-MG

Pollyanna Miranda de Abreu
Analista de Sistemas - Unicentro Newton Paiva
Professora de Comunicação de Dados e Conectividade do CETEL-MG

Consultoria, supervisão e revisão técnica desta edição:

Antonio Pertence Júnior
Engenheiro Eletrônico e de Telecomunicações
Especialista em Processamento de Sinais (Ryerson University – Canadá)
Professor de Telecomunicações da FUMEC (MG)
Professor Titular da Faculdade de Sabará/MG

Reimpressão 2008



2006

This One



553A-GWE-6JSY / righted material

Obra originalmente publicada sob o título
Data Communications and Networking, 3/ed

ISBN 0-07-251584-8

©2004, The McGraw Hill Companies, Inc., 1221 Avenue of the Americas, New York, NY 10020.
All rights reserved.

Capa: *Gustavo Macri*

Supervisão editorial: *Arysinha Jacques Affonso e Denise Weber Nowaczyk*

Editoração eletrônica: *Laser House*

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, 1.091 - Higienópolis
01227-100 São Paulo SP
Fone (11) 3665-1100 Fax (11) 3667-1333

SAC 0800 703-3444

IMPRESSO NO BRASIL
PRINTED IN BRAZIL

Para Faezeh com amor.

Prefácio

As tecnologias de comunicação de dados e as redes de computadores provavelmente são as que mais têm crescido ultimamente. Uma das consequências desse crescimento é o aumento significativo do número de profissões novas, nas quais conhecimentos sólidos sobre tais tecnologias constituem um fator decisivo para o sucesso profissional. Além disso, a quantidade de cursos e de alunos das mais diversas áreas de conhecimento que precisam conhecer muitos dos conceitos sobre comunicação de dados e redes de computadores também tem crescido vertiginosamente.

Características do Livro

Muitas características deste livro-texto foram desenvolvidas de modo a torná-lo particularmente fácil aos estudantes, ajudando-os a compreender os princípios da comunicação de dados e das redes de computadores.

Estrutura

Ao longo do livro, usamos o modelo de cinco camadas da Internet como referencial para o texto, não somente porque a perfeita compreensão do modelo é essencial para a compreensão da teoria das redes atuais, mas porque o modelo de cinco camadas guarda uma estrutura indelével de interdependências: cada camada é montada sobre a camada imediatamente abaixo e suporta a camada de cima. Do mesmo modo, cada conceito introduzido em nosso texto é construído sobre conceitos examinados em seções anteriores. O modelo da Internet foi o escolhido porque é um protocolo totalmente implementado e, aliás, muito bem-sucedido.

Este livro foi escrito para estudantes com pouco ou nenhum conhecimento em telecomunicações ou comunicação de dados. Por este motivo, sempre adotamos uma abordagem que não pressupõe esses pré-requisitos. Na abordagem aplicada ao livro, os estudantes podem aprender primeiramente sobre a comunicação de dados (camadas inferiores) antes de partir para as redes de computadores. Por exemplo, os estudantes irão aprender sobre sinalização, codificação, modulação e detecção de erros antes de aprender sobre a transferência de dados através da Internet. Isso elimina a necessidade de dois cursos: um para comunicação de dados e outro para os conceitos sobre as redes de computadores.

Abordagem Visual

O livro baseia-se em textos e figuras para apresentar temas altamente técnicos sem a utilização de fórmulas complexas. Mais de 700 figuras acompanham as explicações fornecendo material visual e intuitivo embasados nos textos. As figuras são particularmente importantes nas explicações dos

conceitos de redes, principalmente aqueles baseados em conexões e na transmissão. Ambos conceitos são fáceis de serem compreendidos quando tratados visualmente.

Enfatizando Pontos

Destacamos alguns pontos ao longo do texto em caixas de texto como referência rápida sobre um determinado assunto.

Exemplos e Aplicações

Quando julgamos apropriado, incluímos exemplos que ilustram conceitos introduzidos no texto. Eles também auxiliam os estudantes a resolverem os exercícios propostos no final de cada capítulo. Além disso, exemplos de várias aplicações reais são dados ao longo do livro.

Termos-Chave

Cada capítulo possui uma lista de termos-chave para o estudante.

Resumo

Cada final de capítulo traz um resumo do material coberto no capítulo. É dada uma visão geral bastante breve dos pontos importantes do capítulo.

Conjunto Prático

No final de cada capítulo também está incluída uma seção denominada "Pratique os Conhecimentos Adquiridos". O propósito dessa seção é colocar o(a) estudante(a) para trabalhar e salientar alguns conceitos. Ele consiste de três partes: questões de revisão, questões de múltipla escolha e exercícios.

As questões de revisão formam o primeiro nível de teste da compreensão adquirida pelo aluno. As questões de múltipla escolha testam a compreensão que o aluno adquiriu sobre conceitos básicos e terminologia. Os exercícios são testes bem mais elaborados e requerem maior tenacidade dos alunos.

Apêndices

Os apêndices foram planejados para fornecer referência rápida ao aluno ou servir como material de revisão necessário à compreensão dos conceitos discutidos no livro.

Glossário e Acrônimos

O livro traz um glossário extenso e uma lista de acrônimos.

Conteúdos

O livro foi dividido em sete partes. A primeira parte faz um apanhado sobre o que trataremos no livro. A última parte trata a importante questão da segurança de rede. As outras cinco partes foram desenvolvidas para representar as cinco camadas do modelo da Internet. A seguir resumimos os conteúdos em cada parte.

Parte Um: Visão Geral da Comunicação de Dados e das Redes de Computadores

A primeira parte faz um apanhado sobre comunicação de dados e redes de computadores em dois capítulos. O Capítulo 1 cobre os conceitos introdutórios necessários ao resto do livro. O Capítulo 2 introduz o modelo da Internet.

Parte Dois: Camada Física

A segunda parte é uma discussão da camada física no modelo da Internet. Esta parte estende-se dos Capítulos 3 até o 9. Os Capítulos 3 a 6 discutem os aspectos relacionados às telecomunicações na camada física. O Capítulo 7 introduz os meios de transmissão, os quais são controlados pela camada física, apesar de não fazerem parte dela. Os Capítulos 8 e 9 introduzem muitos protocolos relacionados principalmente à camada física.

Parte Três: Camada de Enlace

A terceira parte é toda dedicada à discussão da camada de enlace do modelo da Internet e inclui os Capítulos 10 a 18. O Capítulo 10 é dedicado à detecção de erros. Os Capítulos 11, 12 e 13 discutem as questões relacionadas ao controle do *link* de dados. Os Capítulos 14 a 16 tratam de LANs. As LANs operam nas camadas física e de enlace. O Capítulo 14 trata do padrão de rede mais difundido no mundo: Ethernet. O Capítulo 15 trata das LANs sem fio (WLANs). O Capítulo 16 mostra como podemos conectar LANs para criar redes *backbones*. Os Capítulos 17 e 18 tratam das WANs, outra tecnologia que utiliza as camadas física e de enlace. O Capítulo 17 discute os sistemas de telefonia móvel e de satélite. O Capítulo 18 explica WANs comutadas, como o Frame Relay e ATM.

Parte Quatro: Camada de Rede

A quarta parte discute a camada de rede do modelo da Internet. Esta parte estende-se do Capítulo 19 ao 21. O Capítulo 19 é dedicado aos conceitos e serviços oferecidos na camada de rede e também aborda o roteamento e o processo de *internetworking* da Internet. O Capítulo 20 sobre os protocolos de *internetworking* da Internet. O protocolo principal de *internetworking* da Internet é o protocolo IP (discutido com bastante profundidade). Outros protocolos como ARP, ICMP e IGMP são resumidos para mostrar como eles suportam a operação do protocolo IP. O Capítulo 21 abre a discussão sobre os protocolos de roteamento, tanto em *unicast* quanto em *broadcast*. No roteamento *unicast*, a maioria das vezes é dada ênfase ao problema do *roteamento baseado no vetor de distância, roteamento link state* e *roteamento baseado no vetor de caminhos*. Contudo, RIP, OSPF e BGP são estudados com alguma profundidade, como exemplos de protocolos de roteamento em *unicast*. No roteamento *multicasting*, a ênfase é dada aos métodos de *spanning tree*. Além disso, protocolos como DVMRP, MOSPF, CBT, PIM-DM e PIM-SM são discutidos para mostrar aplicações reais. O IGMP é introduzido para ser o veículo necessário ao roteamento *multicasting*. No final deste capítulo, introduzimos MBONE como um método temporário de *multicast*.

Parte Cinco: Camada de Transporte

A parte cinco é dedicada à discussão da camada de transporte do modelo da Internet e é composta pelos Capítulos 22 e 23. O Capítulo 22 traz uma visão geral da camada de transporte e discute os serviços dessa camada. Ele também introduz dois protocolos da camada de transporte da Internet, UDP e TCP. O Capítulo 23, embora esteja incluído nessa parte do livro, discute duas questões relacionadas não somente à camada de transporte, mas também às duas camadas anteriores: controle de congestionamento e qualidade de serviços. Atualmente, graças às aplicações de multimídia que rodam na Internet, o conhecimento desses tópicos tem sido fundamental para agregar qualidade aos serviços de uma rede.

Parte Seis: Camada de Aplicação

A parte seis é dedicada à discussão da camada de aplicação do modelo da Internet. Esta parte inclui os Capítulos 24 a 28. Os serviços da camada de aplicação são o objetivo do modelo. Todas as outras camadas existem para que os usuários possam acessar as aplicações disponíveis nessa camada. Não tivemos a pretensão de cobrir todas as aplicações existentes para essa camada; de fato, escolhemos alguns exemplos para mostrar os conceitos. O Capítulo 24 define a idéia geral do paradigma cliente-servidor. Nele, introduzimos o conceito da interface *socket* como prelúdio à programação cliente-servidor. O Capítulo 25 cuida do DNS, responsável pelo mapeamento dos endereços da camada de aplicação em endereços da camada rede. O Capítulo 26 trata duas aplicações bastante populares da Internet: *e-mail* e transferência de arquivos. Introduzimos a World Wide Web e o protocolo para acessá-la (HTTP) no Capítulo 27. O Capítulo 28 discute as questões relacionadas às aplicações de multimídia e os problemas envolvidos. Neste capítulo discutimos também alguns protocolos relacionados à telefonia IP (VoIP), teleconferência e *streaming* de áudio/vídeo.

Parte Sete: Segurança de Rede

A parte sete é dedicada às discussões sobre os aspectos ligados à segurança de rede. Atualmente, a segurança não está restrita apenas a uma camada específica do modelo da Internet, ela cobre praticamente todos. Esta parte do livro faz uma breve discussão sobre as idéias principais acerca

da segurança de rede e da informação. Estão incluídos três capítulos nessa parte. O Capítulo 29 trata alguns aspectos relacionados à criptografia. Criptografia de chave secreta (simétrica) e chave pública (assimétrica) são descritas sem muito aprofundamento na teoria dos números. O Capítulo 30 introduz serviços de segurança (confidencialidade, autenticação de usuários, integridade e o não repúdio da informação). Este capítulo também aborda alguns métodos de acesso ao sistema por usuários autenticados. Finalmente, discutimos a importante questão do gerenciamento de chaves usando ambos tipos de criptografia. O Capítulo 31 foca segurança na Internet. Ele explora os protocolos utilizados nas camadas de rede, transporte e aplicação. Além disso, discute os *firewalls* e as VPNs.

Material Suplementar na Web: www.mhhe.com/forouzan*

Online Learning Center

O McGraw-Hill Online Learning Center é um “repositório digital” contendo muitos aspectos pedagógicos do livro e alguns suplementos. À medida que os leitores vão estudando os capítulos, eles podem responder *quizzes online* e verificar a evolução desses estudos. Eles podem também obter informação extra sobre material de leitura como as apresentações em PowerPoint e as figuras animadas de algumas partes do livro. As soluções dos problemas do final dos capítulos também estão disponíveis na Web. Os estudantes têm acesso apenas às soluções dos problemas ímpares, já os professores podem ter acesso irrestrito, através de senha, às soluções dos problemas.

Adicionalmente, a McGraw-Hill torna possível a criação de um *website* dos seus cursos de rede específicos com um produto exclusivo da McGraw-Hill denominado PageOut. Ele não requer nenhum conhecimento prévio de HTML, muitas horas de desenvolvimento e nenhuma habilidade em *webdesign* da sua parte. Em vez disso, o PageOut oferece uma série de *templates* (modelos). Você simplesmente os preenche com a informação do seu curso e clica em um dos 16 projetos. Cada processo de montagem dos conteúdos no *website* leva em torno de 1 hora e, no final, o resultado é um *website* totalmente profissional.

Embora o PageOut ofereça recursos de desenvolvimento *online*, o *website* finalizado oferece ainda uma característica poderosa: um plano de ensino interativo no qual você coloca os conteúdos na mesma seqüência de suas aulas. Assim, quando os estudantes estiverem visitando a sua PageOut, o seu plano de ensino poderá remetê-los ao Online Learning Center (Forouzan) ou a um material específico de sua autoria.

Como Usar o Livro

Este livro foi escrito para público acadêmico e profissional. Os profissionais da área podem utilizá-lo como um guia de estudos para reciclagem dos conhecimentos. Como livro-texto, ele pode ser utilizado em um curso de um semestre. Segue uma sugestão de utilização:

- As partes 1 a 3 são fortemente recomendadas.
- As partes 4 a 6 podem ser adotadas se não houver nenhum curso específico sobre o protocolo TCP/IP na seqüência do curso.
- A parte sete é recomendada se não houver um curso específico sobre segurança de rede na seqüência do curso.

* N. de R. Este site é da editora original (McGraw-Hill), sendo o seu conteúdo em inglês. As apresentações em PowerPoint e as soluções dos exercícios em português podem ser solicitadas para a Bookman editora pelo endereço secretariaeditorial@artimed.com.br. Alunos e professores podem solicitar as apresentações em PowerPoint. As soluções dos exercícios são exclusivamente para professores que adotarem a obra e serão encaminhadas mediante solicitação e comprovação de docência.

Agradecimentos

É óbvio que o desenvolvimento de um livro como este requer o suporte de muitas pessoas. Agradecemos especificamente a Ying-Ping Sarah Liu e Gregory Lee pela incansável assistência na leitura dos manuscritos e na verificação da solução dos problemas dos exercícios.

A contribuição mais importante ao desenvolvimento de um livro vem dos revisores. Não podemos traduzir em palavras nossa profunda gratidão aos muitos revisores que passaram horas lendo e relendo os manuscritos. Foram muitas as excelentes sugestões para melhoria da obra. Gostaríamos de agradecer especialmente às contribuições dos seguintes revisores da terceira edição.

Anthony Barnard, *University of Alabama, Birmingham*
Rayman Meservy, *Brigham Young University*
Scott Campbell, *Miami University*
Arnold C. Meltzer, *George Washington University*
Christophe Veltos, *Minnesota State University, Mankato*
Wenhang Liu, *California State University, Los Angeles*
Sandeep Gupta, *Arizona State University*
Alvin Sek Lee Lim, *Auburn University*
Sherali Zeadally, *Wayne State University*
Tom Hilton, *Utah State University*
Ten-Hwang Lai, *Ohio State University*
Hung Z Ngo, *SUNY, Buffalo*
T. Burrel, *Oklahoma State University*
Hans-Peter Dommel, *Santa Clara University*
Louis Marseille, *Hartford Community College*

Agradecimentos especiais ao pessoal e aos colaboradores da McGraw-Hill. Betsy Jones, nossa editora, provou como uma editora proficiente pode tornar possível o impossível. Emily Lupash, a editora de desenvolvimento, esteve sempre presente quando precisamos dela. Sheila Frank, nossa gerente de projetos, guiou-nos através dos processos de produção com enorme entusiasmo. Além disso, gostaríamos de agradecer a Kara Kudronowicz na produção, Rick Noel no projeto e Patti Scott, na redação.

Notas sobre Marcas Registradas

Ao longo de todo o texto são citadas várias marcas registradas. Em vez de inserir um símbolo de marca registrada (*trademarks ®*) em cada menção sobre uma marca, vamos incluí-las aqui. Em nenhum momento tivemos a intenção de infringir a divulgação de algumas dessas marcas. Outros nomes de produtos e marcas registradas são propriedade dos seus respectivos proprietários.

- Apple, AppleTalk, EtherTalk, LocalTalk e Macintosh são marcas registradas da Apple Computer, INC.
- Bell e StarLan são marcas registradas da AT&T.
- DEC, DECnet, VAX e DNA são marcas registradas da Digital Equipment Corp.
- IBM, SDLC, SNA e IBM PC são marcas registradas da International Business Machines Corp.
- Novell, Netware, IPX e SPX são marcas registradas da Novell, Inc.
- Network File System e NFS são marcas registradas da Sun Micro-systems, Inc.
- PostScript é uma marca registrada da Adobe Systems, Inc.
- UNIX é uma marca registrada da UNIX System Laboratories, Inc, subsidiária da Novell, Inc.
- Xerox é uma marca e Ethernet é uma marca registrada da Xerox Corp.

Copyrighted material

Sumário

PARTE I	VISÃO GERAL DAS COMUNICAÇÕES DE DADOS E DAS REDES DE COMPUTADORES	31
CAPÍTULO 1	INTRODUÇÃO	33
1.1	Comunicação de dados	33
	Componentes	34
	Representação dos Dados	35
	Direção do Fluxo de Dados	36
1.2	Redes	37
	Processamento Distribuído	37
	Critérios de Comparação	37
	Parte Física	38
	Classificação das Redes	42
1.3	A Internet	44
	Uma Breve História	44
	A Internet Hoje	45
1.4	Protocolos e padrões	46
	Protocolos	46
	Padrões	47
	Organizações de Padronização	47
	Padrões da Internet	48
1.5	Termos-chave	49
1.6	Resumo	49
1.7	Pratique os conhecimentos adquiridos	50
	Questões de Revisão	50
	Questões de Múltipla Escolha	50
	Exercícios	51

CAPÍTULO 2 ARQUITETURAS DE REDES	53
2.1 Modelo de camadas	53
Emissor, Receptor e Meio de Transporte	54
Hierarquia	55
Serviços	55
2.2 Modelo de camadas da Internet	55
Processos Peer-to-Peer	56
Funções das Camadas	57
Resumo das Camadas	65
2.3 Modelo OSI	66
2.4 Termos-chave	66
2.5 Resumo	67
2.6 Pratique os conhecimentos adquiridos	67
Questões de Revisão	67
Questões de Múltipla Escolha	67
Exercícios	68
PARTE II CAMADA FÍSICA	69
CAPÍTULO 3 SINAIS	73
3.1 Analógico e digital	73
Informação Analógica e Informação Digital	73
Sinais Analógicos e Sinais Digitais	74
Sinais Periódicos e Sinais Não Periódicos	74
3.2 Sinais analógicos	74
A Onda Senoidal	75
Fase	77
Exemplos de Ondas Senoidais	78
Dominio do Tempo <i>versus</i> Domínio da Frequência	78
Sinais Compostos	79
Largura de Banda	82
3.3 Sinais digitais	85
Intervalo de Sinalização e Número de Bits por Segundo	85
Sinal Digital como um Sinal Analógico Composto	86
Sinal Digital em um Meio Banda Larga	86
Sinal Digital em um Meio de Largura de Banda Limitada	86
Largura de Banda Analógica <i>versus</i> Largura de Banda Digital	88
Altas Taxas de Transmissão em Bits por Segundo	88
3.4 Analógico <i>versus</i> digital	88
Passa-Baixas <i>versus</i> Passa-Banda (ou Passa-Faixa)	89
Transmissão Digital	89
Transmissão Analógica	89
3.5 Limites para a taxa de transmissão de dados	90
Canal Livre de Ruídos: Fórmula para o Número de Bits por Segundo de Nyquist	90
Canal com Ruido: Lei de Shannon	90
Usando Ambos Limites	91
3.6 Transmissão com perdas	92
Atenuação	92

Distorção	93
Ruído	93
3.7 Um pouco mais sobre sinais	94
Throughput	94
Velocidade de Propagação	94
Tempo de Propagação	95
Comprimento de Onda	95
3.8 Termos-chave	96
3.9 Resumo	97
3.10 Pratique os conhecimentos adquiridos	97
Questões de Revisão	97
Questões de Múltipla Escolha	98
Exercícios	100
CAPÍTULO 4 TRANSMISSÃO DIGITAL	103
4.1 Codificação de linha	103
Algumas Características da Codificação de Linha	104
Esquemas de Codificação	106
Outros Esquemas	111
4.2 Codificação de blocos	111
Etapas da Seqüência de Transformação	112
Alguns Blocos de Códigos	113
4.3 Amostragem	114
Pulse Amplitude Modulation (PAM)	115
Pulse Code Modulation (PCM)	116
Taxa de Amostragem: Teorema de Nyquist	117
Quantos Bits por Amostra?	118
Número de Bits por Segundo	118
4.4 Modos de transmissão	119
Transmissão Paralela	119
Transmissão Serial	120
4.5 Termos-chave	123
4.6 Resumo	123
4.7 Pratique os conhecimentos adquiridos	124
Questões de Revisão	124
Questões de Múltipla Escolha	124
Exercícios	126
CAPÍTULO 5 TRANSMISSÃO ANALÓGICA	129
5.1 Modulação de dados digitais	129
Aspectos da Conversão Digital para Analógico	130
Amplitude Shift Keying (ASK)	131
Frequency Shift Keying (FSK)	133
Phase Shift Keying (PSK)	135
Quadrature Amplitude Modulation (QAM)	137
Comparação entre Taxa de Transmissão e Modulação	139
5.2 Modems analógicos	141
Padrões de Modem	142
5.3 Modulação de sinais analógicos	145
Amplitude Modulation – AM	146

Frequency Modulation – FM	148
Phase Modulation – PM	150
5.4 Termos-chave	150
5.5 Resumo	150
5.6 Pratique os conhecimentos adquiridos	151
Questões de Revisão	151
Questões de Múltipla Escolha	151
Exercícios	153
CAPÍTULO 6 MULTIPLEXAÇÃO	155
6.1 FDM	156
Processo de Multiplexação	156
Processo de Demultiplexação	157
A Hierarquia Analógica	158
Outras Aplicações FDM	160
Implementação	160
6.2 WDM	160
6.3 TDM	161
<i>Time Slots e Frames</i>	162
Intercalando Sinais	163
Sincronização	164
Bits de Enchimento	165
Níveis de Sinal Digital (Serviços DS)	166
Hierarquia Digital T	166
TDM Inverso	168
Outras Aplicações da TDM	169
6.4 Termos-chave	169
6.5 Resumo	169
6.6 Pratique os conhecimentos adquiridos	170
Questões de Revisão	170
Questões de Múltipla Escolha	170
Exercícios	171
CAPÍTULO 7 MEIOS DE TRANSMISSÃO	175
7.1 Transmissão guiada	176
Cabo Par Trançado	176
Cabo Coaxial	179
Fibra Óptica	181
7.2 Transmissão sem fios (wireless)	185
Ondas de Rádio	187
Microondas	187
Infravermelho	189
7.3 Termos-chave	190
7.4 Resumo	190
7.5 Pratique os conhecimentos adquiridos	191
Questões de Revisão	191
Questões de Múltipla Escolha	191
Exercícios	193

CAPÍTULO 8 COMUTAÇÃO DE CIRCUITOS E REDES DE TELEFONIA	195
8.1 Comutação de circuitos	195
Comutação por Divisão de Espaço	196
Comutação por Divisão de Tempo	199
TDM Bus	200
Combinações das Comutações por Divisão de Espaço e de Tempo	201
8.2 Redes de telefonia	201
Componentes Macro de uma Rede	202
LATAs	202
Fazendo uma Chamada	204
Serviços Analógicos	205
Serviços Digitais	206
Uma Breve História	207
8.3 Termos-chave	207
8.4 Resumo	208
8.5 Pratique os conhecimentos adquiridos	208
Questões de Revisão	208
Questões de Múltipla Escolha	209
Exercícios	210
CAPÍTULO 9 ACESSO DIGITAL DE ALTA VELOCIDADE: DSL, CABLE MODEMS E SONET	213
9.1 Tecnologia DSL	213
ADSL	213
Outras Tecnologias DSL	215
9.2 Cable modem	216
Redes CATV	216
Redes HFC	217
Compartilhamento	219
CM e CMTS	219
Esquemas de Transmissão de Dados: DOCSIS	220
9.3 SONET	221
Dispositivos SONET	221
Frame SONET	222
Transmissão de Frames	223
STS-1	223
Tributários Virtuais	223
Serviços em Taxas Elevadas	224
9.4 Termos-chave	225
9.5 Resumo	225
9.6 Pratique os conhecimentos adquiridos	226
Questões de Revisão	226
Questões de Múltipla Escolha	226
Exercícios	228
PARTE III CAMADA DE ENLACE DE DADOS	229
CAPÍTULO 10 DETECÇÃO E CORREÇÃO DE ERROS	233
10.1 Tipos de erros	233
Erros Isolados	233

Rajada de Erros	234
10.2 Detecção de erros	235
Redundância	235
Teste da Paridade	236
Cyclic Redundancy Check (CRC)	239
Checksum	242
10.3 Correção de erros	245
Correção de Erros por Retransmissão	245
Correção Antecipada de Erros	245
Correção da Rajada de Erros	248
10.4 Termos-chave	248
10.5 Resumo	249
10.6 Pratique os conhecimentos adquiridos	249
Questões de Revisão	249
Questões de Múltipla Escolha	250
Exercícios	251
CAPÍTULO 11 CONTROLE DO ENLACE DE DADOS E PROTOCOLOS	253
11.1 Controle de fluxo e controle de erro	253
Controle de Fluxo	253
Controle de Erros	254
Mecanismos de Controle de Fluxo e Erros	254
11.2 STOP-AND-WAIT ARQ	254
Operação	255
Transmissão Bidirecional	257
11.3 GO-BACK-N ARO	258
Seqüência de Números	258
Janela de Transmissão	258
Janela de Recepção	259
Variáveis de Controle	259
Relógios	260
Confirmação (ACK)	260
Frames Retransmitidos	260
Operação	260
Tamanho da Janela de Transmissão	261
Transmissão Bidirecional e Piggybacking	262
11.4 Selective repeat ARQ	262
Janelas de Transmissão e Recepção	263
Operação	263
Tamanho da Janela de Transmissão	263
Transmissão Bidirecional e Piggybacking	264
Produto Banda x Tempo de Propagação	265
Pipelining	265
11.5 HDLC	265
Configurações e Modos de Transferência	266
Frames	266
Formato do Frame	266
Tipo de Frame	267
Exemplos	270
Transparência de Dados	271

11.6 Termos-chave	273
11.7 Resumo	274
11.8 Pratique os conhecimentos adquiridos	274
Questões de Revisão	274
Questões de Múltipla Escolha	275
Exercícios	276
CAPÍTULO 12 ACESSO PONTO A PONTO	277
12.1 Protocolo ponto a ponto	277
Formato do Frame	277
Transição de Estados	278
12.2 PPP: pilha de protocolos	279
Link Control Protocol (LCP)	279
Protocolos de Autenticação	281
Network Control Protocol (NCP)	283
Um Exemplo	285
12.3 Termos-chave	286
12.4 Resumo	286
12.5 Pratique os conhecimentos adquiridos	286
Questões de Revisão	286
Questões de Múltipla Escolha	287
Exercícios	288
CAPÍTULO 13 ACESSO MÚLTIPLA	289
13.1 Acesso aleatório	289
Acesso Múltiplo (Multiple Access – MA)	290
Carrier Sense Multiple Access (CSMA)	291
CSMA/CD	293
CSMA/CA	294
13.2 Acesso ordenado	295
Acesso com Reserva	295
Polling	295
Passagem de Permissão (Token-Passing)	296
13.3 Canalização	297
FDMA	297
TDMA	297
CDMA	298
13.4 Termos-chave	302
13.5 Resumo	302
13.6 Pratique os conhecimentos adquiridos	302
Questões de Revisão	302
Questões de Múltipla Escolha	303
Exercícios	304
CAPÍTULO 14 REDES LOCAIS ETHERNET	307
14.1 Ethernet padrão	308
Subcamada MAC	308
Camada Física	310
Implementação da Camada Física	312
Ethernet Interligada por Bridges	314

<u>Switched Ethernet</u>	315
<u>Ethernet Full-Duplex</u>	316
14.2 Fast Ethernet	317
<u>Subcamada MAC</u>	317
<u>Camada Física</u>	317
<u>Implementação da Camada Física</u>	318
14.3 Gigabit Ethernet	321
<u>Subcamada MAC</u>	321
<u>Camada Física</u>	321
<u>Implementação da Camada Física</u>	322
14.4 Termos-chave	324
14.5 Resumo	325
14.6 Pratique os conhecimentos adquiridos	326
<u>Questões de Revisão</u>	326
<u>Questões de Múltipla Escolha</u>	326
<u>Exercícios</u>	327
CAPÍTULO 15 REDES LANs SEM FIO	329
15.1 IEEE 802.11	329
<u>Arquitetura</u>	329
<u>Camada Física</u>	331
<u>Subcamada MAC</u>	332
<u>Mecanismo de Endereçamento</u>	337
15.2 Bluetooth	339
<u>Arquitetura</u>	339
<u>Camadas Bluetooth</u>	340
<u>Camada de Rádio</u>	340
<u>Camada Banda Base</u>	341
<u>L2CAP</u>	344
<u>Outras Camadas Superiores</u>	345
15.3 Termos-chave	345
15.4 Resumo	345
15.5 Pratique os conhecimentos adquiridos	346
<u>Questões de Revisão</u>	346
<u>Questões de Múltipla Escolha</u>	346
<u>Exercícios</u>	348
CAPÍTULO 16 INTERLIGANDO LANs, REDES BACKBONE E LANs VIRTUAIS (VLANs)	349
16.1 Ativos de redes: dispositivos de redes locais	349
<u>Repetidores</u>	349
<u>Hubs</u>	351
<u>Bridges</u>	352
<u>Switch de Camada 2</u>	358
<u>Roteador e Switch de Camada 3</u>	358
16.2 Redes Backbones	358
<u>Backbone Barramento</u>	358
<u>Backbone Estrela</u>	359
<u>Interligando LANs Remotas</u>	360
16.3 LANs virtuais (VLANs)	361
<u>Agrupamento Lógico de Usuários e Recursos</u>	363

Configuração	363
Identificação de VLANs	364
Padrão IEEE 802.1Q	364
Vantagens	364
16.4 Termos-chave	365
16.5 Resumo	365
16.6 Pratique os conhecimentos adquiridos	366
Questões de Revisão	366
Questões de Múltipla Escolha	366
Exercícios	367
CAPÍTULO 17 TELEFONIA CELULAR E REDES DE SATÉLITES	369
17.1 Telefonia celular	369
Princípio de Reuso de Freqüências	369
Transmissão	370
Recepção	371
Handoff	371
Roaming	371
Primeira Geração	371
Segunda Geração	372
Terceira Geração	378
17.2 Redes de satélite	379
Órbitas	380
Footprint	381
Três Categorias de Satélites	381
Satélites GFO	382
Satélites MFO	382
Satélites IFO	384
17.3 Termos-chave	387
17.4 Resumo	387
17.5 Pratique os conhecimentos adquiridos	387
Questões de Revisão	387
Questões de Múltipla Escolha	388
Exercícios	389
CAPÍTULO 18 COMUTAÇÃO DE CIRCUITOS VIRTUAIS: FRAME RELAY E ATM	391
18.1 Comutação de circuitos virtuais	391
Endereçamento Global	392
Identificador de Circuito Virtual	392
Comunicação em Três Fases	392
Fase de Transferência de Dados	393
Fase de Estabelecimento da Conexão	394
Fase de Desconexão do Circuito	396
18.2 Frame Relay	396
Arquitetura	397
Camadas Frame Relay	398
FRADs	400
VOFR	400
LMI	400
Controle de Congestionamento e Qualidade de Serviços	401

18.3 ATM	401
Metas do Projeto ATM	401
Problemas	401
Arquitetura	404
Comutação	406
Camadas ATM	407
Controle de Congestionamento e Qualidade de Serviços	413
LANs ATM	413
18.4 Termos-chave	413
18.5 Resumo	413
18.6 Pratique os conhecimentos adquiridos	414
Questões de Revisão	414
Questões de Múltipla Escolha	415
Exercícios	416
PARTE IV CAMADA DE REDE	419
CAPÍTULO 19 PROCESSOS HOST-TO-HOST: INTERNETWORKING, ENDEREÇAMENTO E ROTEAMENTO	423
19.1 Internetworks	423
Necessidade da Camada de Rede	424
Internet como uma Rede de Comutação de Pacotes	425
Internet como uma Rede sem Conexão	428
19.2 Endereçamento	428
Endereço de Internet	428
Classes de Endereçamento	430
Sub-redes	436
Supernetting	440
Endereçamento sem Classes	440
Configuração Dinâmica de Endereços	441
NAT	443
19.3 Roteamento	446
Técnicas de Roteamento	446
Roteamento Estático versus Roteamento Dinâmico	449
Tabela de Roteamento para as Classes de Endereçamento	450
Tabela de Roteamento para Endereçamento sem Classes	450
19.4 Termos-chave	451
19.5 Resumo	451
19.6 Pratique os conhecimentos adquiridos	452
Questões de Revisão	452
Questões de Múltipla Escolha	453
Exercícios	454
CAPÍTULO 20 PROTOCOLOS DA CAMADA DE REDE: ARP, IPv4, ICMP, IPv6 E ICMPv6	457
20.1 ARP	458
Mapeamento	458
Formato do Pacote ARP	459
Encapsulamento	460
Operação	460

20.2	IP	462
	Datagrama	463
	Fagmentação	466
20.3	ICMP	468
	Tipos de Mensagens	468
20.4	IPv6	471
	Endereços IPv6	471
	Tipos de Endereços	473
	Formato do Parâte IPv6	473
	Fagmentação	474
	ICMPv6	474
	Migração do IPv4 para o IPv6	475
20.5	Termos-chave	477
20.6	Resumo	477
20.7	Pratique os conhecimentos adquiridos	478
	Questões de Revisão	478
	Questões de Múltipla Escolha	478
	Exercícios	480
CAPÍTULO 21 ROTEAMENTO UNICAST E MULTICAST: PROTOCOLOS DE ROTEAMENTO		483
21.1	Roteamento unicast	483
	Métrica	484
	Roteamento Interno e Externo	484
21.2	Protocolos de roteamento unicast	485
	RIP	485
	OSPF	487
	BGP	495
21.3	Roteamento multicast	498
	IGMP	499
	Árvores de Multicast	504
	MBONE	505
21.4	Protocolos de roteamento multicast	506
	DVMRP	506
	MOSPF	508
	CBT	510
	PIM	511
	Aplicações	512
21.5	Termos-chave	512
21.6	Resumo	513
21.7	Pratique os conhecimentos adquiridos	514
	Questões de Revisão	514
	Questões de Múltipla Escolha	514
	Exercícios	518
PARTE V CAMADA DE TRANSPORTE		521
CAPÍTULO 22 PROTOCOLOS DA CAMADA DE TRANSPORTE: TCP E UDP		525
22.1	Comunicação entre processos finais	525
	Paradigma Cliente-Servidor	526
	Mecanismo de Endereçamento	526

<u>Multiplexação e Demultiplexação</u>	528
Serviço sem Conexão <i>versus</i> Serviço Orientado à Conexão	529
Confiável <i>versus</i> não Confiável	531
22.2 User Datagram Protocol (UDP)	532
Números de Portas	532
Datagrama UDP	532
Aplicações	534
22.3 Transmission Control Protocol (TCP)	534
Números de Portas	534
<u>Serviços TCP</u>	<u>535</u>
<u>Numeração de Bytes</u>	<u>537</u>
<u>Número de Seqüência</u>	<u>537</u>
Segmento TCP	538
<u>Conexão</u>	<u>540</u>
<u>Diagrama de Transição de Estados</u>	<u>541</u>
<u>Controle de Fluxo</u>	<u>544</u>
<u>Síndrome da Janela Boba</u>	<u>547</u>
<u>Controle de Erros</u>	<u>548</u>
<u>Relógios do TCP</u>	<u>550</u>
<u>Controle de Congestionamento</u>	<u>552</u>
<u>Outras Características</u>	<u>552</u>
22.4 Termos-chave	553
22.5 Resumo	554
22.6 Pratique os conhecimentos adquiridos	554
<u>Questões de Revisão</u>	<u>554</u>
<u>Questões de Múltipla Escolha</u>	<u>555</u>
<u>Exercícios</u>	<u>557</u>
CAPÍTULO 23 CONTROLE DE CONGESTIONAMENTO E QUALIDADE DE SERVIÇO	559
23.1 Tráfego de dados	559
<u>Descritores do Tráfego</u>	<u>559</u>
<u>Perfis do Tráfego</u>	<u>560</u>
23.2 Congestionamento	561
<u>Performance da Rede</u>	<u>562</u>
23.3 Controle de congestionamento	563
<u>Controle de Congestionamento em Malha Aberta</u>	<u>564</u>
<u>Controle de Congestionamento em Malha Fechada</u>	<u>564</u>
23.4 Dois exemplos	565
<u>Controle de Congestionamento no TCP</u>	<u>565</u>
<u>Controle de Congestionamento no Frame Relay</u>	<u>566</u>
23.5 Qualidade de serviço	568
<u>Características do Fluxo</u>	<u>568</u>
<u>Classes de Fluxo</u>	<u>569</u>
23.6 Técnicas para melhorar a QoS	569
<u>Política de Filas</u>	<u>569</u>
<u>Modelamento do Tráfego</u>	<u>570</u>
Reserva de Recursos	573
Controle de Admissão	573
23.7 Serviços integrados	573
<u>Sinalização</u>	<u>574</u>

Especificações do Fluxo	574
Admissão	574
Classes de Serviços	574
RSVP	574
Problemas com os Serviços Integrados	577
23.8 Serviços diferenciados	577
23.9 QoS em redes comutadas	579
QoS nas Redes Frame Relay	579
QoS nas Redes ATM	580
23.10 Termos-chave	582
23.11 Resumo	583
23.12 Pratique os conhecimentos adquiridos	583
Questões de Revisão	583
Questões de Múltipla Escolha	584
Exercícios	586
PARTE VI CAMADA DE APLICAÇÃO	587
CAPÍTULO 24 ARQUITETURA CLIENTE-SERVIDOR: A INTERFACE SOCKET	591
24.1 Arquitetura cliente-servidor	591
Relacionamento	591
Concorrência	593
Processos	594
24.2 A interface Socket	594
Sockets	594
Servidor Iterativo sem Conexão	596
Servidor Orientado à Conexão	597
24.3 Termos-chave	598
24.4 Resumo	599
24.5 Pratique os conhecimentos adquiridos	599
Questões de Revisão	599
Questões de Múltipla Escolha	599
Exercícios	601
CAPÍTULO 25 DOMAIN NAME SYSTEM (DNS)	603
25.1 Espaço de nomes	603
Espaço de Nomes Plano	604
Espaço de Nomes Hierárquico	604
25.2 Espaço de nomes de domínio	604
Componentes do Nome de Nível Superior	604
Nome do Domínio	605
Domínio	606
25.3 Distribuição do espaço de nomes	606
Hierarquia de Servidores de Nomes	606
Zonas	607
Servidor Raiz	608
Servidores Primários e Secundários	608
25.4 DNS na Internet	608
Domínios Genéricos	609

Domínios Geográficos	609
Domínios de Reserva	610
25.5 Resolvendo nomes	611
Resolver	611
Mapeando Nomes em Endereços	611
Mapeando Endereços IP em Nomes	612
Resolução Recursiva	612
Resolução Iterativa	612
Caching	613
25.6 Mensagens DNS	613
Cabeçalho	614
Seção Pergunta	614
Seção Resposta	614
Seção Autoridade	615
Seção Informação Adicional	615
25.7 DDNS	615
25.8 Encapsulamento	615
25.9 Termos-chave	616
25.10 Resumo	616
25.11 Pratique os conhecimentos adquiridos	617
Questões de Revisão	617
Questões de Múltipla Escolha	617
Exercícios	618
CAPÍTULO 26 CORREIO ELETRÔNICO (SMTP) E TRANSFERÊNCIA DE ARQUIVO (FTP)	619
26.1 Correio eletrônico	619
Recebendo Correio	620
Endereços	620
User Agent (UA)	620
MIME	622
Mail Transfer Agent (MAT)	627
Entrega de e-mail	628
Protocolos de Acesso à Caixa de Correio	629
E-mail Baseado na Web	630
26.2 Transferência de arquivos	631
Conexão	632
Comunicação	632
Transferência do Arquivo	634
Interface do Usuário	635
FTP Anônimo	636
26.3 Termos-chave	636
26.4 Resumo	636
26.5 Pratique os conhecimentos adquiridos	637
Questões de Revisão	637
Questões de Múltipla Escolha	637
Exercícios	639
CAPÍTULO 27 HTTP e WWW	641
27.1 HTTP	641
Transação HTTP	642

<u>Mensagem Pedido</u>	642
<u>Mensagem Resposta</u>	644
<u>Cabeçalhos das Mensagens</u>	645
<u>Alguns Exemplos</u>	646
<u>Algumas Características Adicionais</u>	646
27.2 World Wide Web (WWW)	648
<u>Hipertexto e Hipermídia</u>	648
<u>Arquitetura de um Browser</u>	649
<u>Documentos Estáticos</u>	649
HTML	650
Exemplos	651
Documentos Dinâmicos	653
Common Gateway Interface (CGI)	653
<u>Exemplos</u>	654
<u>Documentos Ativos</u>	655
Java	656
<u>Exemplos</u>	658
27.3 Termos-chave	659
27.4 Resumo	660
27.5 Pratique os conhecimentos adquiridos	660
Questões de Revisão	660
Questões de Múltipla Escolha	660
<u>Exercícios</u>	663
CAPÍTULO 28 MULTIMÍDIA	665
28.1 Digitalizando áudio e vídeo	666
<u>Áudio Digitalizado</u>	666
<u>Video Digitalizado</u>	666
28.2 Compressão de áudio e vídeo	667
<u>Compressão de Áudio</u>	667
<u>Compressão de Vídeo</u>	668
28.3 Streaming de áudio e vídeo armazenado	672
<u>Primeira Abordagem: Usando um Servidor de Web</u>	672
Segunda Abordagem: Usando um Servidor de Web com Metafile	673
Terceira Abordagem: Usando um Servidor de Mídia	673
Quarta Abordagem: Usando um Servidor de Mídia e o RTSP	674
28.4 Streaming de áudio e vídeo em tempo real	675
28.5 Streaming de áudio e vídeo interativo	675
Características	675
Protocolo de Transporte em Tempo Real	679
<u>Real-Time Transport Control Protocol (RTCP)</u>	680
28.6 Voz sobre IP (VoIP)	681
<u>SIP</u>	681
<u>H.323</u>	683
28.7 Termos-chave	685
28.8 Resumo	685
28.9 Pratique os conhecimentos adquiridos	686
Questões de Revisão	686
Questões de Múltipla Escolha	686
<u>Exercícios</u>	688

PARTE VII SEGURANÇA	689
CAPÍTULO 29 CRIPTOGRAFIA	693
29.1 Introdução	693
29.2 Criptografia com chave simétrica	694
Tipos Básicos de Cifras	695
Cifra por Blocos	698
Modos de Operação	701
29.3 Criptografia com chave pública	703
RSA	705
Escolhendo as Chaves Pública e Privada	706
29.4 Termos-chave	706
29.5 Resumo	706
29.6 Pratique os conhecimentos adquiridos	707
Questões de Revisão	707
Questões de Múltipla Escolha	707
Exercícios	708
CAPÍTULO 30 SEGURANÇA DA INFORMAÇÃO, AUTENTICAÇÃO DE USUÁRIOS E GERENCIAMENTO DE CHAVES	711
30.1 Segurança da informação	711
Privacidade	711
Autenticação da Mensagem	713
Integridade	713
O Problema do Não Repúdio	713
30.2 Assinatura digital	713
Assinando um Documento	713
Assinando a Síntese de um Documento	714
30.3 Autenticação do usuário	716
Autenticação do Usuário Através da Criptografia com Chave Simétrica	716
Autenticação do Usuário Através da Criptografia com Chave Pública	718
30.4 Gerenciamento de chaves	718
Distribuição da Chave Simétrica	718
Certificação com Chave Pública	724
30.5 Kerberos	726
Servidores	726
Operação	727
Usando Servidores Diferentes	728
Kerberos Versão 5	728
Realms (Domínios de Autenticação)	728
30.6 Termos-chave	729
30.7 Resumo	729
30.8 Pratique os conhecimentos adquiridos	730
Questões de Revisão	730
Questões de Múltipla Escolha	730
Exercícios	731
CAPÍTULO 31 PROTOCOLOS DE SEGURANÇA NA INTERNET	733
31.1 Segurança na camada de rede: IPSec	733
Security Association	734

Dois Modos de Operação	734
Dois Protocolos de Segurança	734
Protocolo ESP	736
31.2 Segurança na camada de transporte	737
Posição do TLS	738
Dois Protocolos	738
31.3 Segurança na camada de aplicação: PGP	739
31.4 Firewalls	740
Filtros de Pacotes	741
Proxy Firewall	741
31.5 Virtual Private Network	742
Redes Privadas	742
Implementando Privacidade	743
Tecnologia VPN	745
31.6 Termos-chave	746
31.5 Resumo	746
31.6 Pratique os conhecimentos adquiridos	746
Questões de Revisão	746
Questões de Múltipla Escolha	747
Exercícios	748
APÊNDICE A CÓDIGO ASCII	749
APÊNDICE B SISTEMAS DE NUMERAÇÃO E CONVERSÃO ENTRE BASES	753
B.1 Sistemas de numeração	753
Números Decimais	754
Números Binários	754
Números Octais	755
Números Hexadecimais	756
B.2 Conversão entre bases	757
Convertendo de Outros Sistemas para Decimal	757
Convertendo de Decimal para Outros Sistemas	758
De Binário para Octal ou Hexadecimal	759
De Octal para Hexadecimal ou Binário	759
APÊNDICE C O MODELO OSI	760
C.1 O modelo	760
C.2 Camadas do modelo OSI	760
As Quatro Primeiras Camadas	760
Camada de Sessão	761
Camada de Apresentação	761
Camada de Aplicação	762
C.3 Comparação	763
APÊNDICE D CÓDIGO 8B/6T	765
APÊNDICE E CÁLCULO DO CHECKSUM	768
E.1 Notação binária	768
Soma Parcial	769
Sum	769
Checksum	769

E.2 Notação hexadecimal	769
Soma Parcial	769
Sum	770
Checksum	770
APÊNDICE F ESTRUTURA DE UM ROTEADOR	771
F.1 Componentes	771
Portas de Entrada	771
Portas de Saída	772
Processador de Roteamento	772
Circuitos de Comutação	772
APÊNDICE G LANs ATM	775
G.1 Arquitetura das LANs ATM	775
Arquitetura ATM Pura	776
Arquitetura ATM Legada	776
Arquitetura Mista	777
G.2 Emulação de LANs (LANE)	777
G.3 Modelo cliente-servidor	778
LAN Emulation Client (LEC)	778
LAN Emulation Configuration Server (LECS)	778
Broadcast/Unknown Server (BUS)	779
G.4 Arquitetura mista cliente-servidor	779
APÊNDICE H PROGRAMAS CLIENTE-SERVIDOR	780
H.1 Programas cliente-servidor UDP	780
Programa Servidor UDP	780
Programa Cliente UDP	781
H.2 Programas cliente-servidor TCP	782
Programa Servidor TCP	782
Programa Cliente TCP	784
APÊNDICE I RFCs	785
APÊNDICE J PORTAS UDP E TCP	787
APÊNDICE K ENDEREÇOS DE CONTATO	789
Lista de Acrônimos	791
Glossário	795
Índice	821

VISÃO GERAL DAS COMUNICAÇÕES DE DADOS E DAS REDES DE COMPUTADORES

As comunicações de dados e as redes de computadores são objetos que saíram do universo tecnológico e caíram no domínio público. Produtos como os aparelhos de MP3 e telefones celulares não são mais restritos ao mundo de magia da alta tecnologia, mas sim brinquedos para todos, desde pré-adolescentes aos avós. O progresso na tecnologia de comunicação de dados e nas redes de computadores está acontecendo numa velocidade assustadora. Para se ter uma idéia, as antenas de televisão tipo *bunny-ear* (antena interna de TV) são coisas pré-históricas, se comparadas às transmissões digitais a cabo e via satélite dos dias atuais. Hoje os escritórios são movidos por conexões *wireless*. Para o usuário final dessas tecnologias, o único requisito é saber como utilizá-las, isto é, ter o *know how*. Contudo, um estudante destes campos do conhecimento humano deve estar familiarizado com tópicos e conceitos mostrados na Figura 1.

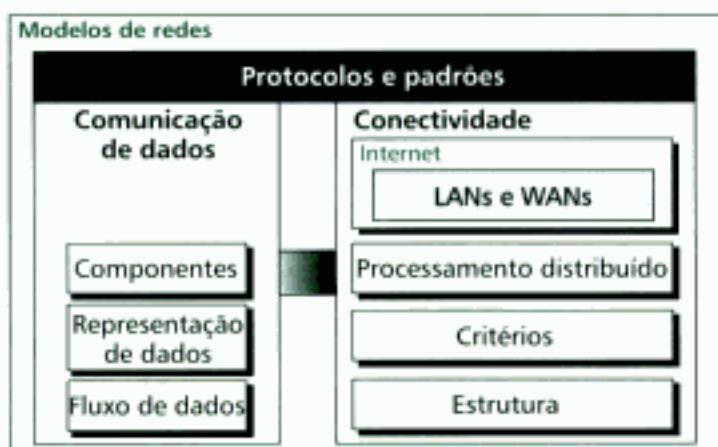


Figura 1 Visão geral.

Comunicação de Dados

As redes de comunicação existem para que dados possam ser enviados de um lugar para outro, essa é a idéia básica da comunicação de dados. Para que esse assunto seja entendido completamente,

te, devemos compreender os componentes físicos de uma rede; saber representar os diferentes tipos de dados e estarmos aptos a criar/gerenciar um fluxo de dados.

Conectividade

A comunicação de dados entre locais remotos pode ser realizada através de um processo denominado *conectividade*, que envolve desde a conexão de computadores, meios e dispositivos de redes (os ativos de redes). Assim, quando estivermos falando sobre conectividade, queremos que se tenha em mente três conceitos: processamento distribuído, critérios (protocolos) de redes e infra-estrutura de redes.

Rede Locais e Redes Geograficamente Distribuídas

Redes de computadores são classificadas em duas categorias principais: redes locais (LANs) – *Local Area Networks* – e as redes geograficamente distribuídas (WANs) – *Wide Area Networks*. Essas redes têm diferentes tipos de características e funcionalidades. Em geral, uma conexão LAN é uma coleção de computadores e dispositivos periféricos numa região limitada, tal como um prédio ou um campus. Uma LAN quase sempre está sob o domínio privado de uma empresa. Já uma rede WAN é uma coleção de LANs e estende-se geograficamente por enorme região.

Internet

A Internet, principal foco deste livro, é uma coleção de LANs e WANs unidas por dispositivos de *internetworking*. Na Figura 1, mostramos esse relacionamento na caixa intitulada *Internet* que encerra as LANs e WANs. Entretanto, a Internet é mais do que simplesmente uma conexão física de LANs e WANs; ela também é um emaranhado de protocolos e padrões de *internetworking*.

Protocolos e Padrões

Os protocolos e padrões são vitais para a implementação de uma rede de comunicação de dados. Os protocolos referem-se às regras; já um padrão é um protocolo adotado por organismos internacionais de padronização e empresas do ramo. No diagrama da Figura 1, a caixa *Protocolos e Padrões* abrange tanto o conjunto comunicação de dados quanto *networking* (redes) para enfatizar que cada área ou competência tem uma regra própria.

Modelos de rede

Os modelos de rede servem para organizar, unificar e controlar os componentes de *hardware* e *software* da rede de comunicação de dados. Embora o termo “Modelos de rede” pareça estar relacionado apenas às redes, ele também se aplica à comunicação de dados em si.

Capítulos

No Capítulo 1, discutimos brevemente os três primeiros tópicos – comunicação de dados, *networking* (redes) e os padrões. Os modelos de rede, alicerces fundamentais para o restante do livro, são descritos no Capítulo 2.

Introdução

Hoje em dia as redes de comunicação de dados mudaram nosso modo de fazer negócios e nosso estilo de vida. A tomada de uma decisão de negócios tem sido feita cada vez mais rapidamente e aqueles que as tomam requerem cada vez mais informações concretas (confiáveis). Por que esperar uma semana para que um relatório originado na Alemanha chegue pelo correio, se ele pode aparecer quase que instantaneamente através de uma rede de computadores? Mas antes de nos perguntarmos quão rapidamente podemos consegui-lo numa transmissão, precisamos conhecer como as redes funcionam, quais os tipos de tecnologias disponíveis e qual projeto melhor atende às nossas necessidades.

O desenvolvimento do computador pessoal modificou tremendamente os negócios, a indústria, a ciência e a educação. Uma revolução semelhante está acontecendo nas redes de comunicação de dados. Tecnologias avançadas estão tornando possível transmitir cada vez mais sinal e em velocidades cada vez maiores. Como resultado, os serviços estão evoluindo para permitirem o uso a essa capacidade estendida, incluindo a extensão para estabelecer serviços tais como um *conference calling*, chamada em espera, mensagens de voz e identificador de chamada.

O fato básico é: as redes de comunicação de dados ainda estão na infância. O objetivo é ser possível trocar informação em tempo hábil, como textos, áudio e vídeo a qualquer lugar do mundo. Queremos acessar a Internet rápida e confiavelmente, a qualquer momento, e fazer *downloads* e/ou *uploads* da informação contida nos *sites* sem muita demora.

Este capítulo foca quatro pontos fundamentais: comunicação de dados, redes, a Internet e os protocolos/padrões. De início, discutiremos amplamente a definição de comunicação de dados. Então, definiremos redes como uma via rápida (*highway*) por onde os dados podem viajar. Em seguida, discutiremos a Internet como um bom exemplo de uma *internetworking* (i.e., uma rede de redes). Finalmente, discutiremos os diferentes tipos de protocolos, a diferença entre protocolos e padrões e as organizações que recomendam um determinado conjunto de padrões.

1.1 COMUNICAÇÃO DE DADOS

Quando comunicamos, compartilhamos informação. Este compartilhamento pode ser local ou remoto. Em geral, entre indivíduos, a comunicação local acontece face a face, enquanto que a comunicação remota toma lugar a longas distâncias. A palavra **telecomunicações** quer dizer “comunicação a longas distâncias” (do grego *tele* = longe, ao longe, distante) e inclui a telefonia, telegrafia e a televisão.

O termo **dados** refere-se à informação apresentada em qualquer forma onde concordem as partes, a que originou (criou) e a que fará uso dos dados.

Comunicação de dados é a troca de informação entre dois dispositivos através de alguma forma de meio de comunicação, por exemplo um par de fios. Para que a comunicação de dados aconteça, os dispositivos de comunicação devem ser parte de um sistema de comunicações feito a partir da combinação *hardware* (equipamento físico) e *software* (programas). A eficiência de um sistema de comunicação de dados depende fundamentalmente de três características:

1. **Entrega (delivery).** O sistema deve entregar os dados ao destino correto. Os dados devem ser recebidos somente pelo dispositivo ou usuário de destino.
2. **Confiabilidade.** O sistema deve garantir a entrega dos dados. Dados modificados ou corrompidos numa transmissão são inúteis.
3. **Tempo de Atraso.** O sistema deve entregar dados em um tempo finito e predeterminado. Dados entregues tarde são pouco úteis. Por exemplo, no caso de transmissões de áudio e de vídeo, os atrasos não são desejáveis, de modo que eles devem ser entregues praticamente no mesmo instante em que foram produzidos, isto é, sem atrasos significativos. Este tipo de entrega é denominada *transmissão em tempo real*.

Componentes

Um sistema básico de comunicação de dados é composto de cinco elementos (veja a Figura 1.1).

1. **Mensagem.** A **mensagem** é a informação (dados) a ser transmitida. Pode ser constituída de texto, números, figuras, áudio ou vídeo – ou qualquer combinação desses.
2. **Transmissor.** O **transmissor** é o dispositivo que envia a mensagem de dados. Pode ser um computador, uma estação de trabalho (*workstation*), um telefone, uma câmera de vídeo e assim por diante.
3. **Receptor.** O **receptor** é o dispositivo que recebe a mensagem. Pode ser um computador, uma estação de trabalho, um telefone, uma câmera de vídeo e assim por diante.
4. **Meio.** O **meio de transmissão** é o caminho físico por onde viaja uma mensagem originada no transmissor e dirigida ao receptor. Pode ser um par trançado, cabo coaxial, fibra óptica ou ondas de rádio (microondas terrestre ou via satélite).
5. **Protocolo.** Um **protocolo** é um conjunto de regras que governa a comunicação de dados. Ele representa um acordo entre os dispositivos que se comunicam. Sem um protocolo, dois dispositivos podem estar conectados, mas sem comunicação entre si. Por exemplo, uma pessoa que fala apenas o francês dificilmente compreenderá o que diz outra pessoa que só fala o japonês.

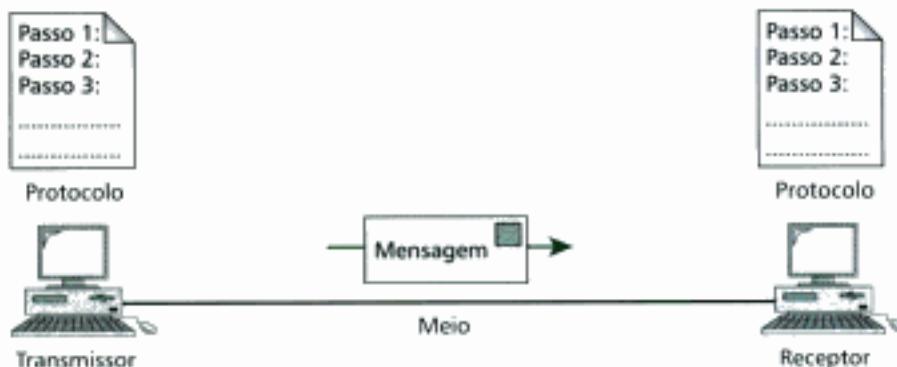


Figura 1.1 Cinco componentes da comunicação de dados.

Representação dos Dados

Hoje em dia a informação se apresenta de diferentes formas, tais como caracteres numéricos ou alfanuméricos, visual ou audível.

Caracteres

Em comunicação de dados, um caractere é representado por um padrão ou uma seqüência de *bits*. Os e 1s. O número de *bits* no padrão depende do número de símbolos na linguagem ou código. Por exemplo, na escrita inglesa existem 26 símbolos (A, B, C,..., Z) para representar as letras maiúsculas, 26 símbolos (a, b, c,..., z) para representar as letras minúsculas, 10 símbolos (0,1,2,..., 9) para representar caracteres numéricos e vários símbolos (.,?:;!,!) para representar a pontuação. Outros símbolos tais como espaço, recuo e o *tab* são usados para alinhamento e formatação de textos.

Foram desenvolvidos diferentes conjuntos de padrões de *bits* para representar os tipos mais diversos de caracteres. Cada conjunto é denominado **código** e o processo de representação de símbolos é chamado **codificação**.

ASCII A American National Standards Institute (ANSI) desenvolveu um código denominado *American Standard Code for Information Interchange* (ASCII). Este código utiliza 7 *bits* para representar cada símbolo. Isto significa que 128 (2^7) símbolos diferentes podem ser definidos por esse código. O padrão de *bits* do código ASCII completo está apresentado no Apêndice A.

ASCII Estendido Para ajustar o tamanho de cada padrão a 1 *byte* (8 *bits*), foi adicionado ao código ASCII um *bit* 0 à esquerda do algarismo mais significativo. Desse modo, cada padrão passou a ocupar exatamente um *byte* de memória. Em outras palavras, no código ASCII estendido, o primeiro padrão é 00000000 e o último é 01111111.

Unicode Todos os códigos anteriores foram criados para representar símbolos da língua inglesa. Nenhum deles é capaz de representar símbolos em outras línguas. Para isso é necessário um código de grande capacidade de representação. De uma união entre fabricantes de *hardware* e *software* surgiu um código denominado Unicode que se utiliza de 16 *bits* e é capaz de representar até 65.536 (2^{16}) símbolos. Seções diferentes desse código são alocadas para símbolos em diferentes línguas do mundo. Algumas partes do código são deixadas para símbolos gráficos e/ou símbolos especiais.

ISO A Organização Internacional de Padronização (*International Organization for Standardization*), conhecida simplesmente por ISO, desenvolveu um código com um padrão de 32 *bits*. Este código representa cerca 4.294.967.296 (2^{32}) símbolos e é suficiente para representar qualquer símbolo no mundo.

Numéricos

Números também são representados através de um padrão de *bits*. Entretanto, um código como o ASCII não é utilizado para representar números; um número geralmente é convertido para binário sem nenhuma representação adicional. O motivo principal é que isso simplifica as operações matemáticas a serem aplicadas nos números. O Apêndice B lista o sistema binário e as equivalências com os demais sistemas.

Imagens

Atualmente, as **imagens** também são representadas por um padrão de *bits*. Porém, o mecanismo de representação é diferente. Na forma mais simples, uma imagem é dividida numa matriz de *pixels*, onde cada *pixel* representa um pequeno ponto. O tamanho do *pixel* depende de uma propriedade do elemento gráfico denominada *resolução*. Por exemplo, uma imagem pode ser dividida em 1000 *pixels* ou 10.000 *pixels*. No segundo caso, a imagem possui uma representação melhor, mais definida, ou de maior resolução. O preço que se paga por uma resolução melhor é um aumento significativo na quantidade de memória necessária ao armazenamento da figura.

Após a divisão em *pixels*, cada *pixel* é atribuído a um padrão de *bits*. O tamanho e o valor do padrão depende da imagem que se deseja representar. Para uma imagem formada de pontos em

preto-e-branco (p. ex., tabuleiro de xadrez), o padrão de um único *bit* (*1-bit*) é suficiente para representar um pixel.

Para representar imagens coloridas, cada pixel colorido é decomposto em três cores primárias (básicas): vermelho, verde e azul (RGB). Assim, a intensidade de cada cor é medida e um padrão de *bits* (usualmente 8 *bits*) lhe é atribuído. Em outras palavras, cada pixel possui três padrões de *bits*: um para representar a intensidade da cor vermelha, outro para representar a intensidade da cor verde e mais para representar a intensidade da cor azul.

Áudio

Áudio é uma representação para o som. O áudio tem uma natureza diferente dos caracteres, números ou imagens. Ele é contínuo, não discreto. Até mesmo quando utilizamos um microfone para converter um sinal sonoro ou musical para um sinal elétrico, nós criamos um sinal contínuo. Veremos, nos Capítulos 4 e 5, como converter um sinal de áudio para digital ou outro sinal analógico.

Vídeo

Vídeo pode ser produzido como um sinal contínuo (p. ex., por uma câmera de TV) ou pode ser uma combinação de imagens, cada qual uma sequência discreta, montadas para gerar a idéia de movimento. Novamente, nos Capítulos 4 e 5, veremos como converter um sinal de vídeo para digital ou outro sinal analógico.

Direção do Fluxo de Dados

Uma comunicação entre dois dispositivos pode acontecer de três maneiras diferentes: *simplex*, *half-duplex* ou *full-duplex*.

Simplex

No **modo simplex**, a comunicação é unidirecional, como numa rua de mão única. Somente um dos dois dispositivos no link é capaz de transmitir; logo o outro só será capaz de receber (veja a Fig. 1.2).

Teclados e monitores comuns de computador são dois bons exemplos de dispositivos *simplex*. O teclado é um dispositivo essencialmente de entrada e o monitor um dispositivo de saída.

Half-Duplex

No **modo half-duplex**, cada estação pode transmitir e receber, mas nunca ao mesmo tempo. Quando um dos dispositivos está transmitindo o outro está recebendo e vice-versa (veja Fig. 1.3).

O modo *half-duplex* funciona como uma via de uma única pista bidirecional. Enquanto os carros trafegam em uma direção, os carros na direção oposta devem esperar pela liberação da via. Numa transmissão *half-duplex*, toda a capacidade do canal é dada ao dispositivo que estiver transmitindo no momento. Os exemplos incluem os *walkie-talkies* e aos rádio tipo CBs (*Citizens Band*).

Full-Duplex

No **modo full-duplex** (também chamado de **duplex**), ambas estações podem transmitir e receber simultaneamente (veja Fig. 1.4).



Figura 1.2 Simplex.



Figura 1.3 Half-duplex.

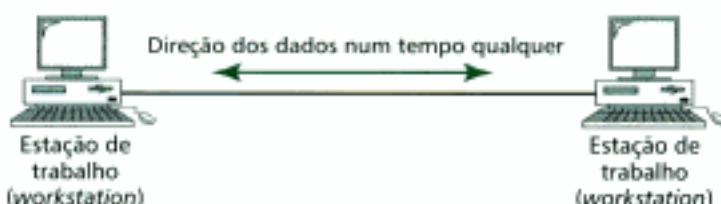


Figura 1.4 Full-duplex.

O modo *full-duplex* é semelhante a uma via de mão dupla, isto é, aquela cujo tráfego flui nas duas direções ao mesmo tempo. No modo *full-duplex*, sinais em direções opostas compartilham a capacidade do *link* ou canal. Esse compartilhamento pode acontecer de duas formas: o *link* possui dois caminhos físicos de transmissão distintos (separados), um para enviar e o outro para receber; a capacidade do canal é dividida entre os sinais viajando em direções opostas.

Um exemplo típico de comunicação *full-duplex* é o canal de voz da rede telefônica. Quando duas pessoas estão se comunicando através do telefone, ambas podem ouvir e falar ao mesmo tempo.

1.2 REDES

Uma **rede** é um conjunto de dispositivos conectados por *links* de comunicação (denominados freqüentemente de *nós*). Um nó pode ser um computador, uma impressora ou qualquer outro dispositivo capaz de enviar e/ou receber dados gerados outros nós da rede.

Processamento Distribuído

Hoje em dia, a maioria das redes usam **processamento distribuído** para executar uma tarefa entre muitos computadores (tipicamente PCs e estações de trabalho – *workstations*). Isso é muito mais eficiente que entregar todo o poder de processamento a uma única máquina poderosa e deixá-la responsável por todos os aspectos computacionais da rede.

Critérios de Comparação

Redes podem ser comparadas segundo alguns critérios de comparação. Os critérios mais importantes são a *performance*, a confiabilidade e a segurança.

Performance

A *performance* de uma rede pode ser medida de diferentes formas, dentre elas incluem-se o tempo de trânsito e o tempo de resposta. O tempo de trânsito é o intervalo de tempo necessário para uma mensagem viajar de um dispositivo a outro. O tempo de resposta é o tempo decorrido entre uma solicitação e uma resposta. A *performance* de uma rede depende de inúmeros outros fatores, tais como o número de usuários, o meio de transmissão, a capacidade do *hardware* conectado à rede e a eficiência do *software* que roda na rede.

Confiabilidade

Além da garantia de entrega, a **confiabilidade** de uma rede é medida pela freqüência de falhas, o tempo de reconfiguração de *link* após uma falha e a robustez da rede numa catástrofe.

Segurança

Segurança de rede é um critério cuja finalidade é assegurar a proteção dos dados e das informações que trafegam na rede do acesso não autorizado.

Parte Física

Antes de discutir as redes, precisamos definir alguns atributos de redes.

Tipo de Conexão

Uma rede é constituída de dois ou mais dispositivos juntos através de *links*. Um *link* é um caminho de comunicação por onde são transferidos dados de um dispositivo a outro. Pictoricamente, é mais simples imaginar qualquer *link* como sendo uma linha desenhada entre dois pontos. Para que a comunicação aconteça, dois dispositivos devem estar conectados a um mesmo *link* ao mesmo tempo. Há duas formas possíveis de conexão: ponto a ponto e multiponto.

Ponto a Ponto Uma conexão **ponto a ponto** proporciona um *link* dedicado entre dois dispositivos. Toda a capacidade do *link* é reservada para a comunicação entre esses dois dispositivos. A maioria das conexões ponto a ponto se utilizam de um cabo para conectar os dois dispositivos, mas existem outras opções como um *link* de microondas e de satélite (veja a Fig. 1.5). Quando você muda o canal de TV por um controle remoto infravermelho, você está estabelecendo uma conexão ponto a ponto entre o controle remoto e o sistema de controle da TV.

Multiponto Uma conexão **multiponto** (*multinode* ou *multidrop*) é aquela na qual mais de dois dispositivos compartilham um único *link* (veja Fig. 1.6).

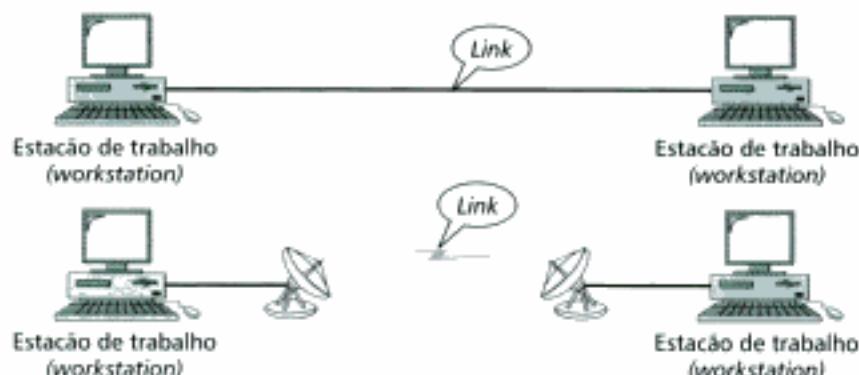


Figura 1.5 Conexão ponto a ponto.



Figura 1.6 Conexão multiponto.

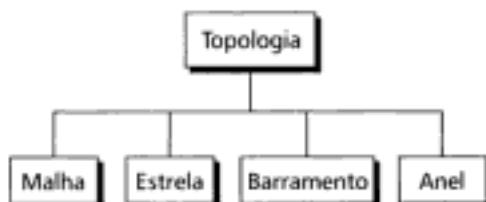


Figura 1.7 Tipos de topologias.

Num ambiente multiponto, a capacidade do canal é compartilhada, espacial ou temporalmente, entre os dispositivos do *link*. O *compartilhamento espacial* é caracterizado pela utilização simultânea do *link* de comunicação. Se os usuários compartilham o *link* mediante um revezamento, a conexão é do tipo *compartilhamento temporal*.

Topologia Física

O termo **topologia física** refere-se ao modo segundo o qual uma rede é montada fisicamente. Dois ou mais dispositivos formam um *link*; dois ou mais *links* geram uma topologia de rede. A topologia de uma rede é a representação geométrica do relacionamento entre todos os *links* e dispositivos conectados uns aos outros (usualmente os **nós**). Existem quatro topologias básicas: malha, estrela, barramento e anel (veja a Fig. 1.7).

Malha Numa **topologia em malha** cada dispositivo possui um *link* dedicado com os demais dispositivos da rede. O termo *dedicado* significa que o tráfego no *link* fica restrito ao dois dispositivos que estiverem se comunicando. Numa malha totalmente conectada existem $n(n - 1)/2$ canais físicos interligando n dispositivos. Para suportar tantos *links*, cada dispositivo na rede deve possuir $n - 1$ interfaces de entrada/saída (E/S – veja Fig. 1.8).

A topologia em malha apresenta muitas vantagens quando comparada às demais. Primeiramente, a utilização de *links* dedicados possibilita o tráfego dos dados apenas na conexão que estiver fechada. Isso elimina os problemas de tráfego decorrentes da necessidade de compartilhar o *link* entre muitos dispositivos. Além disso, uma topologia em malha é robusta. Se um *link* tornar-se indisponível, não ocorre a incapacitação de comunicação no sistema como um todo. Mais uma vantagem associada à malha é a privacidade ou segurança. Qualquer comunicação que viaje ao longo da linha dedicada estará disponível apenas para os dispositivos conectados ao *link*. A fronteira física topológica evita que usuários externos a ela obtenham acesso à informação ali transmitida. Finalmente, os *links* ponto a ponto facilitam a identificação e isolamento de falhas. Com isso, o tráfego pode ser desviado para evitar problemas nos *links* suspeitos. Isto ajuda ao gerente ou suporte de rede a localizar precisamente a falha. Logo, facilita a detecção da causa e a tomada de decisão para apontar uma solução para o problema.

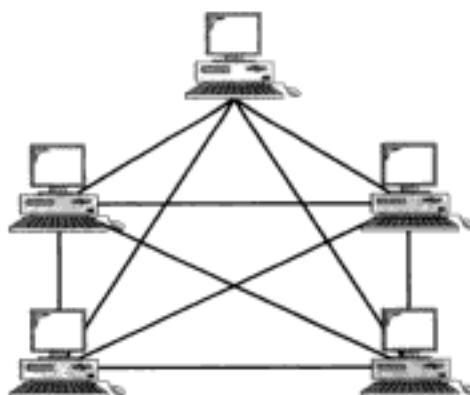


Figura 1.8 Topologia em malha totalmente conectada (para cinco dispositivos).

As principais desvantagens de uma rede em malha estão relacionadas ao cabeamento excessivo e à quantidade de interfaces E/S necessárias ao funcionamento da rede. A primeira desvantagem deve-se ao fato de que cada dispositivo precisa ser conectado aos demais na rede. Isto torna a instalação e configuração da rede bastante difícil. Ainda em relação ao cabeamento, o sistema de canaletas para acomodar os cabos pode tornar-se maior que o espaço disponível no ambiente de rede (nas paredes, tetos ou pisos). Finalmente, o custo do *hardware* exigido para conectar cada *link* (interfaces E/S e cabos) pode tornar-se proibitivamente elevado. Por essas razões, a topologia em malha, quando implementada, apresenta-se de maneira bastante limitada – por exemplo, como um *backbone* interligando os computadores principais (p. ex., servidores) de um rede híbrida formada de diversas outras topologias.

Estrela Numa **topologia em estrela**, cada dispositivo comunica-se dedicadamente a um controlador ou concentrador no centro da estrutura. Este concentrador freqüentemente é denominado **hub***. Assim, os dispositivos não são conectados diretamente uns aos outros. Diferentemente da topologia em malha, não há comunicação direta de um dispositivo para outro numa topologia em estrela. O concentrador age como um elemento intermediário no processo de comunicação entre dois dispositivos: se um dispositivo quer enviar dados a outro, primeiramente envia os dados para o concentrador que, por sua vez, replica os dados para o dispositivo de destino (veja Fig. 1.9).

O custo de uma topologia em estrela é mais acessível do que da topologia em malha. Numa topologia em estrela, cada dispositivo necessita somente de um *link* e uma interface E/S para conectar-lo aos demais da rede. Isto facilita a instalar e a reconfigurar toda a rede. Além do mais, a quantidade de cabos exigidos na montagem da rede em estrela é muito menor, se comparada à topologia em malha. Isto porque cada dispositivo é conectado ao concentrador por um, e apenas um, cabo.

Outras vantagens incluem a robustez da topologia. Se um *link* falha, apenas ele é afetado. Todos os demais permanecem ativos. Este fator também contribui para tornar mais fácil a identificação e o isolamento da falha. Uma vez que, colocado em funcionamento, o *hub* pode ser utilizado para monitorar problemas e evitar *links* defeituosos.

Entretanto, embora a topologia em estrela exija menos cabeamento que a topologia em malha, cada nó deve estar interligado a um *hub* central. Por esse motivo, essa topologia requer mais cabos que algumas outras topologias (tal como em anel e barramento).

Barramento Todos os exemplos de topologias anteriores descrevem conexões ponto a ponto. Uma **topologia em barramento** é diferente, ela prevê conexões multiponto. Um cabo longo funciona como um **backbone (espinha dorsal)** interconectando todos os dispositivos numa rede (veja Fig. 1.10).

Os nós são conectados ao *backbone* através de pequenos segmentos de cabos e conectores de pressão (*taps*). O segmento de cabo faz a conexão entre o dispositivo e o cabo principal. Um *tap* é um conector que permite estender o comprimento de um cabo principal até o dispositivo que se deseja conectar ao meio. Como os sinais de comunicação viajam ao longo do *backbone*, parte da energia que eles transportam é transformada em calor. Desse modo, à medida que viajam mais e mais ao longo do comprimento do cabo, vão sendo enfraquecidos pela dissipação de potência do sinal sob a forma de calor. Isto limita o número e a distância mínima entre os *taps* que um barramento pode suportar.

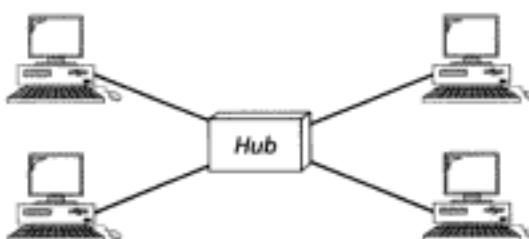


Figura 1.9 Topologia estrela.

* N. de R. T.: Frequentemente, pode-se encontrar um *switch* ou roteador como elementos concentradores numa topologia estrela.

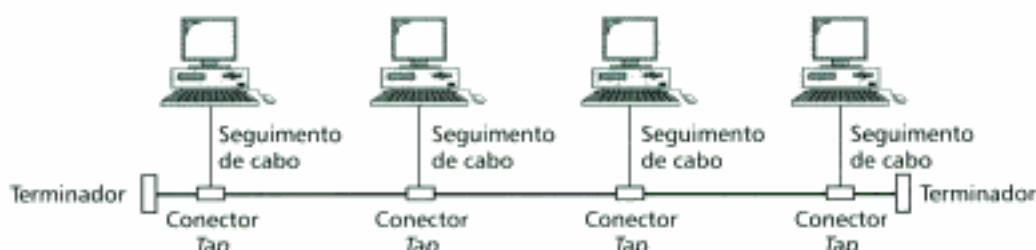


Figura 1.10 Topologia barramento.

A maior vantagem de uma topologia em barramento é a facilidade de instalação. O cabo *backbone* pode ficar situado ao longo de um caminho mais eficiente, então conectar os nós através de segmentos de cabo de vários comprimentos possíveis. Desse modo, a topologia em barramento usa menos cabeamento que as topologias em malha ou em estrela. Numa topologia em estrela, por exemplo, quatro dispositivos numa mesma sala usam quatro segmentos de cabos para alcançar o *hub central**. Num barramento esta redundância é eliminada. Um único cabo *backbone* lançado no ambiente de rede é todo o recurso necessário à interligação dos dispositivos. Cada segmento de cabo para interligar os dispositivos precisa apenas atingir o ponto mais próximo possível do *backbone*.

Dentre as desvantagens desse tipo de rede estão incluídas a dificuldade de reconexão e o isolamento de uma falha. Uma rede em barramento é projetada para otimizar o processo de instalação da rede. Por isso, muitas vezes torna-se difícil adicionar novos pontos de rede no ambiente. Além disso, a reflexão dos sinais nos *taps* degradam a qualidade do sinal no cabo *backbone*. Esta degradação pode ser controlada limitando o número e espaçando convenientemente os dispositivos a serem conectados num certo comprimento de cabo. Adicionar novos dispositivos pode assim requerer a modificação ou substituição de todo o *backbone*.

Por fim, uma falha ou desconexão no cabo do barramento para qualquer tipo de transmissão, até mesmo entre os dispositivos que não estão próximos ao segmento onde se encontra o problema. A parte danificada do cabo reflete os sinais de volta em todas as direções, gerando ruídos de ambos os lados.

Anel Numa **topologia em anel** cada dispositivo possui uma conexão ponto a ponto (dedicada) somente com os dois dispositivos mais próximos dele. Um sinal é transmitido ao longo do anel numa única direção, de um dispositivo a outro, até alcançar o destino. Cada dispositivo no anel incorpora um repetidor. Quando um dispositivo no anel recebe um sinal endereçado a outro dispositivo, o repetidor regenera o sinal de dados e o transmite adiante (veja Fig. 1.11).

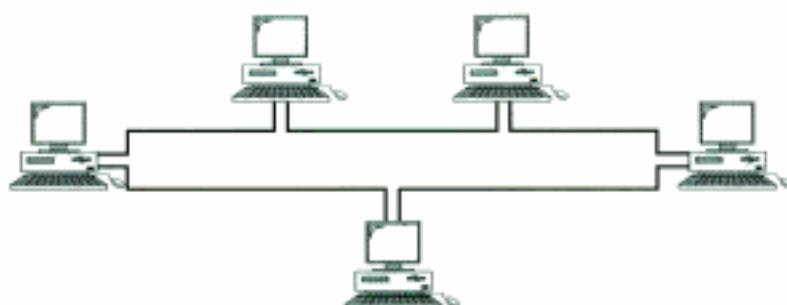


Figura 1.11 Topologia em anel.

* N. de R. T.: Vale a pena mencionar que, em geral, o tipo de cabo utilizado numa topologia em estrela é diferente do tipo de cabo numa topologia em barramento. Em estrela é utilizado frequentemente o par trançado e em barramento o cabo coaxial.

Um anel é relativamente fácil de se instalar e reconfigurar. Cada dispositivo é interligado somente com os dois vizinhos imediatos (física ou logicamente). Em termos de conexão, para acrescentar ou retirar dispositivos nessa rede são necessárias somente duas modificações. Os únicos vínculos que se deve observar são o meio físico e o tráfego (comprimento máximo do cabo e número de dispositivos). Além de que, o isolamento de uma falha nesse tipo de rede é bastante simples. Geralmente, um sinal está sendo transmitido a todo instante no anel. É gerado um alerta se qualquer dos dispositivos não receber um sinal dentro de um período de tempo predeterminado. O alerta informa ao operador da rede que existe um problema e onde ele está localizado.

Entretanto, o tráfego unidirecional pode ser uma enorme desvantagem. Num anel simples, uma quebra (tal como a desconexão de uma estação) pode desabilitar toda a rede. Este inconveniente pode ser resolvido através da adoção de um anel duplo ou de um chaveamento capaz de redirecionar as conexões endereçadas ao ponto de quebra.

Classificação das Redes

Hoje em dia, quando falamos em redes, geralmente nos referimos aos três tipos básicos: rede local, rede metropolitana e rede geograficamente distribuída. Dentro de cada uma dessas classificações, cada rede é determinada pelo tamanho, pelo tipo de domínio, pela distância geográfica que ela cobre e pela arquitetura física (veja Fig. 1.12).

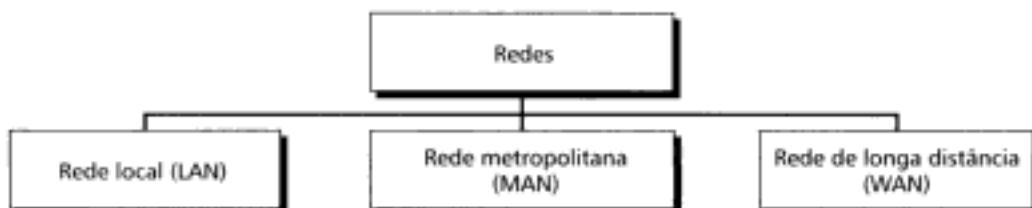


Figura 1.12 Classificação de redes.

Rede Local (LAN) Uma **rede de área local** (Local Area Network – LAN) é administrada privativamente e os *links* entre dispositivos estão localizados dentro de uma sala, escritório, edifício ou campus (veja Fig. 1.13). Uma LAN pode ser formada por dois PCs e uma impressora dentro de um escritório particular ou por centenas de dispositivos numa empresa, incluindo periféricos de áudio e vídeo. Uma LAN depende essencialmente da infra-estrutura de uma organização ou de uma empresa e do tipo de tecnologia utilizada. Atualmente, o tamanho aceitável para uma LAN está limitado a poucos quilômetros.

As LANs são projetadas para permitirem o compartilhamento de recursos entre computadores pessoais ou estações de trabalho. Ainda, os recursos compartilhados podem incluir *hardware* (impressora, gravadora de CD, etc.), *software* (programas aplicativos) ou dados. Um exemplo comum de uma LAN, encontrado em muitos ambientes de trabalho, interliga computadores dentro de um mesmo grupo de trabalho, por exemplo, estações de trabalho da engenharia ou PCs da contabilidade. Um dos computadores da LAN, geralmente aquele de grande capacidade de processamento e de armazenamento de informações, pode ser configurado para tornar-se um servidor da rede e utilizado na autenticação de todos os grupos de trabalho na LAN. *Software's* podem ser instalados nesse servidor central e serem disponibilizados para todos aqueles que necessitarem acessá-lo dentro da LAN. Nesse exemplo, o tamanho da LAN pode ser determinado pelas restrições ao número de usuários por cópia do *software* ou por restrições ao número de usuários licenciados para acessar o sistema operacional.

Além do tamanho da rede, o tipo de meio de transmissão e a topologia são outros mecanismos que distinguem as LANs dos demais tipos de redes. Em geral, uma dada LAN usa somente um tipo de meio de transmissão. As topologias mais comuns para LANs são barramento, anel e estrela.

Tradicionalmente, as LANs transferem dados a velocidades de 4 a 16 megabits por segundo (Mbps). Atualmente, porém, as velocidades estão aumentando e, em muitos ambientes, as LANs já

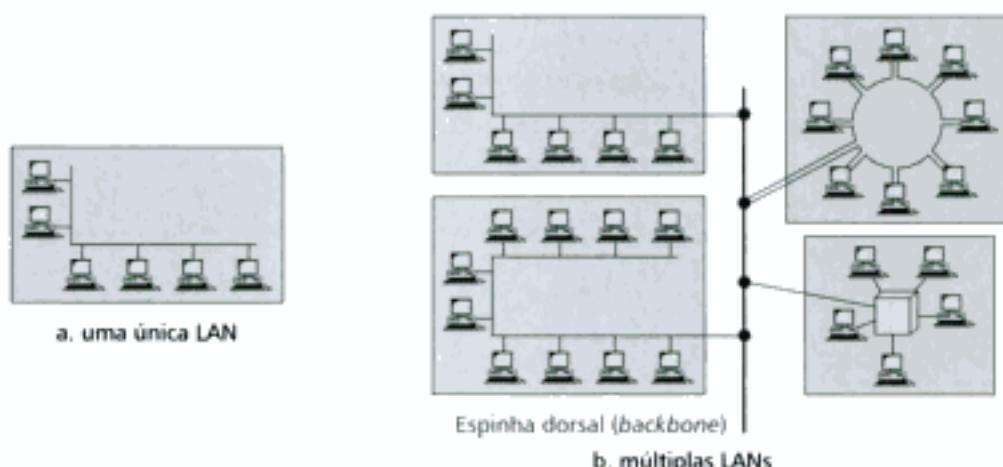


Figura 1.13 Rede local (LAN).

operam a 100Mbps, sendo freqüentes as empresas que já estudam e testam as LANs padronizadas em velocidades da ordem de *gigabits* por segundo (Gbps). As LANs são discutidas em profundidade nos Capítulos 14, 15 e 16.

Redes Metropolitanas Uma **rede de área metropolitana** (Metropolitan Area Network – MAN) é projetada para se estender por toda uma cidade. Pode ser constituída de uma única rede, tal como uma rede de TV a cabo, ou pode conectar muitas LANs entre si, formando uma rede maior, de tal maneira que os recursos possam ser compartilhados de LAN para LAN ou de dispositivo para dispositivo. Por exemplo, uma empresa pode utilizar uma MAN para conectar as LANs de todos os escritórios distribuídos numa cidade (veja Fig. 1.14).

Uma MAN pode ser totalmente administrada por uma empresa privada ou pode ser provida por uma empresa pública, tal como uma companhia telefônica. Muitas empresas telefônicas disponibilizam uma MAN de serviços bastante popular denominado Serviço de Dados sem Conexão de Alta Velocidade (Switched Multi-Megabit Data Services – SMDS).

Rede Geograficamente Distribuída (WAN) Uma **rede de longa distância** (Wide Area Network – WAN) proporciona a transmissão de dados, voz, imagem e vídeo a grandes distâncias geográficas podendo compreender um país, um continente ou até mesmo todo o mundo (veja Fig. 1.15).

Diferentemente das LANs (às quais depende do próprio *hardware* para transmissão), as WANs podem utilizar as redes públicas, redes sob concessão ou alugadas, equipamentos privados de comunicação ou combinações desses para atingir uma distância praticamente ilimitada na superfície do planeta.

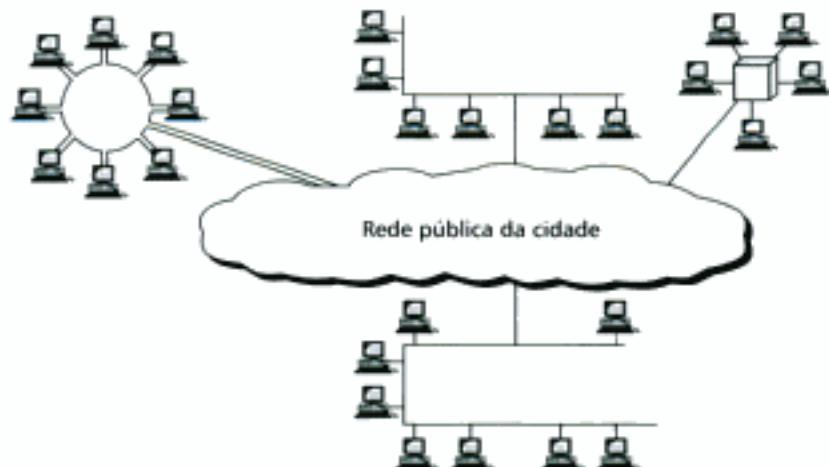


Figura 1.14 MAN.



Figura 1.15 Rede de longa distância (WAN).

Uma WAN sob domínio de uma única empresa é denominada *rede corporativa*. As WANs são discutidas nos Capítulos 17 e 18.

Internetworks Quando duas redes ou mais são conectadas entre si, elas se tornam uma **internetwork** ou **internet** (observe que a letra i é escrita em letra minúscula).

1.3 A INTERNET

A Internet tem revolucionado em muitos aspectos nosso modo de vida. Ela afetou desde o modo de fechar negócios empresariais até nosso modo de passar as horas vagas. Experimente contar de quantas formas você tem utilizado a Internet ultimamente. Talvez você a esteja utilizando para trocar correio eletrônico (*e-mail*) com um sócio, pagando uma conta, lendo um jornal de uma outra cidade ou olhando a programação dos cinemas locais. Ou talvez você possa estar pesquisando algum tópico na área de medicina, fazendo reservas num hotel, participando de um *chat* com um colega distante (*Trekker*) ou comparando os preços de automóveis. A Internet é um sistema de comunicação que colocou o poder da informação ao alcance dos dedos e a organizou para nosso uso.

A Internet é um sistema organizado. Iniciaremos contando uma breve história da Internet. Através dela, daremos uma descrição do que se tornou a Intenet dos dias de hoje.

Uma Breve História

Vimos que uma **rede** é um grupo de dispositivos conectados, tais como computadores e impressoras. Uma **internet** (note a letra i minúscula) são duas redes ou mais redes que podem se comunicar. A internet mais notável é a nossa famigerada **Internet** (letra I maiúscula), composta de centenas de milhares de redes interconectadas. A Internet é utilizada tanto por indivíduos quanto organizações como agências governamentais, escolas, centros de pesquisa, corporações e bibliotecas em mais de 100 países. Milhões de pessoas são usuários dela. Este extraordinário sistema de comunicação teve origem nos idos de 1969.

Na metade da década de 60, os *mainframes* (computadores de grande porte) dentro de organizações de pesquisa eram dispositivos de processamento isolados. Computadores de diferentes marcas eram incapazes de se comunicar. A Advanced Research Projects Agency (ARPA) agência do departamento de defesa dos Estados Unidos (Departament of Defense – DoD) estava interessada em encontrar um modo de conectar os computadores de tal modo que os pesquisadores dela pudessem compartilhar pesquisas, reduzindo custos e evitando a duplicação de esforços.

Em 1967, num encontro da Association Computing Machinery (ACM) o grupo do ARPA apresentou as idéias para a **ARPANET**, uma rede pequena de computadores. A idéia central era de que cada computador (*host*), não necessariamente do mesmo fabricante, pudesse se conectar a um computador específico, denominado *interface message processor* (IMP). Os IMPs, por sua vez, tinham a capacidade de se conectar e de se comunicar entre si, assim como estabelecer comunicação com o computador *host* que pedia acesso à rede.

Em 1969 a ARPANET tornou-se uma realidade. Foram fechados quatro nós, na University of California (UCLA) em Los Angeles, na University of California em Santa Barbara (UCSB), em Stanford Research Institute (SRI) e na University of Utah via IMPs para formar uma rede. Um *software* batizado de *Network Control Protocol* (NCP) controlou a comunicação entre os *hosts*.

Em 1972, Vint Cerf e Bob Kahn, ambos haviam feito parte do grupo da ARPANET, colaboraram entre si no conhecido *Internetting Project*. Num artigo de 1973, eles estabeleceram protocolos muito bem estruturados para promover entrega de pacotes de dados de uma ponta a outra numa rede. Este artigo sobre o protocolo de controle de transmissão (Transmission Control Protocol – TCP) incluía conceitos como encapsulamento, o datagrama e as funções de um *gateway*.

Pouco tempo depois, autoridades da área da computação decidiram dividir o TCP em dois protocolos: o **Transmission Control Protocol (TCP)** e o **Internetworking Protocol (IP)**. O IP seria o responsável pelo roteamento do datagrama enquanto o TCP assumiria as funções de alto nível como segmentação, reagrupamento e detecção de erros. O protocolo de *internetworking* tornou-se conhecido como TCP/IP.

A Internet Hoje

A Internet sofreu muitas modificações desde a década de 60. A Internet hoje não é mais uma simples estrutura hierárquica. Ela é constituída de muitas LANs e WANs trabalhando juntas, conectando dispositivos e chaveando estações. É difícil fazer uma representação exata da Internet porque ela está modificando continuamente – novas redes estão sendo agregadas, as redes atuais estão expandindo o número de endereços existentes, redes de empresas extintas ou falidas estão sendo removidas, etc. Hoje em dia, a maioria dos usuários que querem estabelecer uma conexão com a Internet usam os serviços de acesso dos provedores de Internet (Internet Service Provider – ISPs). Existem provedores de acesso que operam nos planos mundial, nacional, regional ou local. A Internet hoje é disponibilizada por empresas privadas e não governamentais. A Figura 1.16 apresenta uma visão conceitual (não geográfica) da Internet.

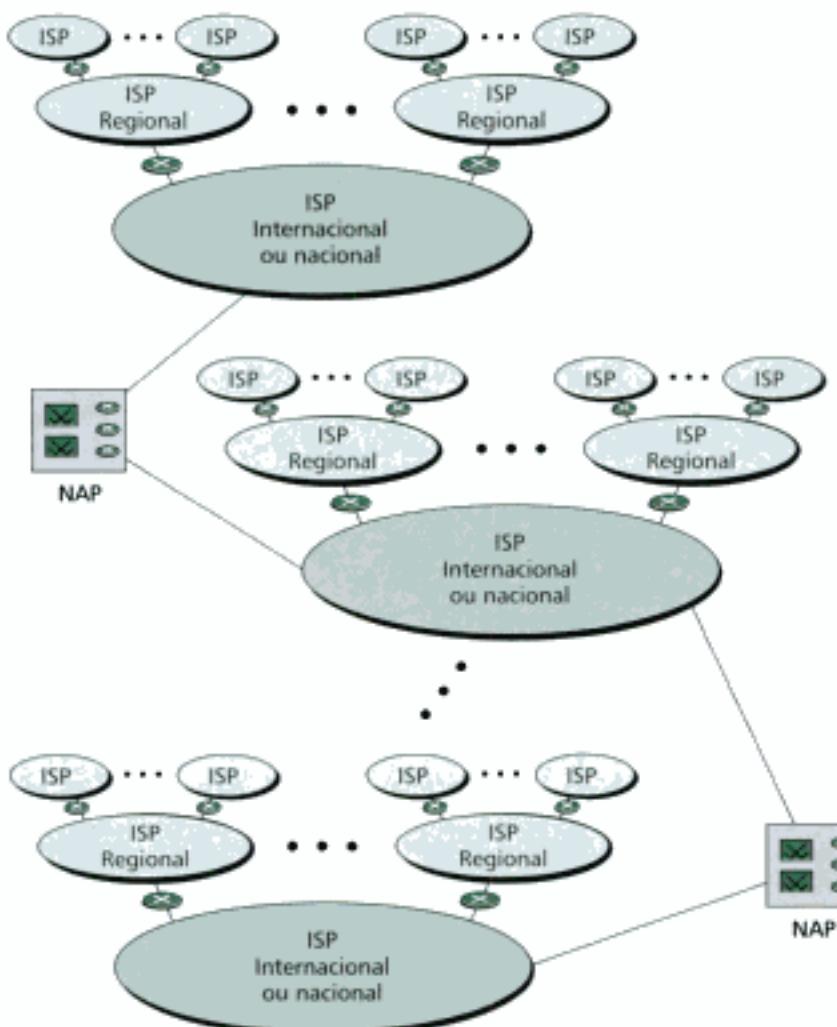


Figura 1.16 Internet hoje.

Provedor Internacional de Acesso

No topo da hierarquia da Internet estão os provedores de serviços de acesso internacionais que se encarregam de conectar nações.

Provedor Nacional de Acesso (National Service Provider – NSP)

Os NSPs são redes tipo *backbones* criadas e mantidas por empresas especializadas. Há muitas empresas desse tipo operando na América do Norte; dentre as mais conhecidas estão SprintLink, PSINet, UUNet Technology, AGIS e internet MCI. Para assegurar a conectividade entre usuários finais, estas redes *backbones* mantêm-se conectadas por complexas centrais de chaveamento denominadas **Pontos de acesso à rede** (Network Access Points – NAPs). Algumas redes NSP também são conectadas umas às outras através de centrais de chaveamento privadas chamadas *peering points*. Os NSPs normalmente operam em velocidades de transmissão muito altas (acima de 600Mbps).

Provedor Regional de Acesso (Regional Internet Service Providers)

Os provedores regionais de acesso ou **ISP regional** são os menores ISPs que podem ser conectados a um ou mais NSP. Eles formam o terceiro nível com menor velocidade de acesso na hierarquia.

Provedor Local de Acesso a Internet (Local Internet Service Provider)

Um provedor local proporciona acesso direto à Internet aos usuários finais. Os ISPs locais podem se conectar aos ISPs regionais ou, então, se conectar diretamente a um ou mais NSP. A maioria dos usuários finais estão conectados a algum ISP local. Note que um ISP local pode ser uma empresa prestadora de serviços de acesso à Internet, uma corporação que proporciona serviços de acesso aos próprios empregados ou uma organização sem fins lucrativos, tais como escolas ou universidades, que administra a própria rede. Cada um desses ISPs pode estabelecer conexão com ISP regional ou nacional.

1.4 PROTOCOLOS E PADRÕES

Nesta seção, definiremos dois conceitos largamente utilizados no jargão de redes: protocolos e padrões. Primeiramente, definiremos o conceito de *protocolo*, que é sinônimo de regra. Em seguida, discutiremos os *padrões*, que são normas sobre a utilização das regras.

Protocolos

Em redes de computadores ocorre comunicação entre entidades em diferentes sistemas. Entende-se por **entidade** qualquer dispositivo capaz de enviar ou receber informação. Entretanto, duas entidades não podem simplesmente trocar um fluxo de dados e esperar que a informação seja compreendida. Para que a comunicação seja estabelecida, as entidades devem concordar acerca do protocolo utilizado. Um **protocolo** é um conjunto de regras que governa a comunicação de dados. Um protocolo define o que é comunicado, de que forma é comunicado e quando será comunicado. Os elementos chave de um protocolo são a sintaxe, a semântica e a temporização (*timing*).

- **Sintaxe.** A sintaxe refere-se à estrutura ou ao formato dos dados e à ordem segundo a qual os dados são apresentados. Por exemplo, um protocolo simples poderia especificar que o primeiro *byte* indicasse o endereço da origem, o segundo *byte* indicasse o endereço de destino e o resto do fluxo de dados fosse a mensagem ou informação propriamente dita.
- **Semântica.** A semântica revela qual o significado de cada conjunto ou seção de *bits*. Então, a semântica define como um padrão particular será interpretado e que ação será tomada baseada nessa interpretação? Por exemplo, um endereço identifica uma rota a ser seguida no roteador ou o endereço final da mensagem?

- **Temporização.** A temporização ou *timing* está ligada a duas características: quando os dados devem ser enviados e quanto rápido eles podemos enviá-los. Por exemplo, se uma fonte de dados produzir uma massa de dados a 100Mbps mas o destino puder receber apenas a 1Mbps, a transmissão sobrecarregará o receptor e todos os dados serão praticamente perdidos.

Padrões

Padrões são essenciais na criação e manutenção de mercados abertos e competitivos para fabricantes de equipamentos, na garantia da interoperabilidade de dados, nacional e internacional, e na tecnologia das telecomunicações e dos processos. Eles formam a via para que fabricantes, comerciantes, agências governamentais e outros provedores de serviços assegurem o tipo de interconectividade necessária aos mercados atuais e comunicações em nível internacional. Os padrões em comunicações de dados estão divididos em duas categorias: padrões *de facto* e *de jure*.

- **De facto.** Padrões que ainda não foram aprovados por um corpo ou comitê organizado, mas têm sido muito difundidos e adotados como padrão. Os **padrões de facto** são frequentemente estabelecidos e impostos por fabricantes de equipamentos que procuram definir a funcionalidade de um novo produto ou tecnologia.
- **De jure.** Padrões reconhecidos por um corpo ou comitê organizado formam os **padrões de jure**.

Organizações de Padronização

Padrões nascem da cooperação entre os comitês de criação de padrões, fóruns e em agências de regulamentação dos governos.

Comitês de Criação de Padrões

Embora existam muitas organizações que se dedicam à criação e ao estabelecimento de padrões, a comunicação de dados na América do Norte* apóia-se primeiramente nos padrões publicados pelas seguintes organizações:

- **International Organization for Standardization (ISO).** A ISO é formada por um corpo internacional cujos membros, em maior número, fazem parte dos comitês de criação de padrões dos vários países que compõem e aceitam a ISO. A ISO é bastante ativa no desenvolvimento de cooperação com os domínios da ciência, da tecnologia e da atividade econômica.
- **International Telecommunication Union – Telecommunication Standards Sector (ITU-T).** No início da década de 70, um certo número de países iniciaram um processo de definição de um padrão nacional para as telecomunicações, mas havia, como era de se esperar, muita incompatibilidade entre os padrões. Coube a Organização das Nações Unidas a responsabilidade de formação, como parte constituinte da ITU, de um comitê (o **Consultative Committee for International Telegraphy and Telephony – CCITT**). Este comitê era voltado à pesquisa e ao estabelecimento de padrões para as telecomunicações em geral, telefonia e sistemas de comunicação de dados. A partir de março de 1993, este comitê passou a ser chamado de International Telecommunication Union – Telecommunication Standards Sector (ITU-T).
- **American National Standards Institute (ANSI).** A despeito do nome, a American National Standards Institute é uma organização totalmente privada, sem fins lucrativos e sem vínculos com o governo dos Estados Unidos. Todavia, todas as atividades da ANSI são reconhecidas e contam com o apoio do governo americano, sendo que os cargos na ANSI são de importância primária (vital) para o país.

* N. de R. T.: Cabe lembrar que, no Brasil, a organização de padronização é a ABNT (Associação Brasileira de Normas Técnicas).

- **Institute of Electrical and Electronics Engineers (IEEE).** O Institute of Electrical and Electronics Engineers é a maior sociedade profissional de engenheiros no mundo. Internacional em escopo, ela ajuda no avanço da teoria, da criatividade e da qualidade dos produtos nos campos da engenharia elétrica e eletrônica, assim como todos os braços relacionados à engenharia. Como uma meta, o IEEE supervisiona o desenvolvimento e a adoção de padrões internacionais para a computação e as comunicações.
- **Electronics Industries Association (EIA).** Alinhada com a ANSI, a Electronic Industries Association é uma organização sem fins lucrativos dedicada à promoção de qualquer item relacionado aos produtos eletrônicos. Dentre as atividades desenvolvidas por ela estão a educação/divulgação junto ao público e os esforços (*lobby*) junto ao governo para adoção de padrões da indústria. No campo da teoria da informação, a EIA tem feito contribuições significativas para definição das interfaces físicas e as especificações elétricas de sinais para a comunicação de dados.

Fóruns

O desenvolvimento de tecnologia nas telecomunicações avança muito mais rapidamente que os comitês de padronização são capazes de ratificar os padrões. Os comitês de padronização seguem procedimentos rígidos e, por natureza, funcionam lentamente. Para se adaptar à necessidade de trabalhar os modelos, agilizar os acordos e facilitar o processo de padronização, muitos grupos especializados têm desenvolvido *fóruns* constituídos de representantes das corporações interessadas. Os fóruns trabalham junto às universidades e aos usuários para testar, avaliar e padronizar novas tecnologias. Concentrando os esforços numa tecnologia particular, os fóruns são capazes de agilizar a aceitação e o uso destas tecnologias na comunidade das telecomunicações. Os fóruns apresentam as conclusões dos respectivos trabalhos aos comitês de padronização.

Agências Reguladoras

Toda e qualquer tecnologia de comunicação está sujeita à regulamentação pelas agências governamentais tais como a **Federal Communications Commission (FCC)** nos Estados Unidos. O propósito dessas agências é proteger o interesse público regulamentando as comunicações de rádio, televisão e a cabo. A FCC tem autoridade sobre o comércio nacional (EUA) e internacional quando o assunto são as comunicações.

Padrões da Internet

Padrões da Internet são especificações úteis, exaustivamente testadas, voltadas para quem trabalha com a Internet. Regulamentam formalmente o que deve ser seguido em termos de Internet. Existe um procedimento rigoroso para conceder a uma especificação o *status* de padrão da Internet. Uma proposta de especificação começa no nível de Internet *draft* (minuta). O **Internet draft** é um documento de trabalho (um trabalho em progresso) sem *status* oficial e tempo de vigência de seis meses. Esse é o tempo para que as autoridades competentes julguem o documento. De acordo com a recomendação dessas autoridades da Internet, um *draft* pode se publicado como um **Request for Comment (RFC)**. Cada RFC é editado, atribuído um número de identificação e colocado à disposição de quem se interessar. Os RFCs abrangem vários níveis da hierarquia da Internet e são classificados de acordo com o requisito de cada nível.

1.5 TERMOS-CHAVE

Advanced Research Projects Agency (ARPA)	Modo <i>half-duplex</i>
American National Standards Institute (ANSI)	Modo <i>simplex</i>
ARPANET	Nível de publicação
Áudio	Nó
<i>Backbone</i>	Padrão da Internet
Código	Padrões <i>de facto</i>
Comunicação de dados	Padrões <i>de jure</i>
Conexão multiponto	Performance
Conexão ponto a ponto	Processamento Distribuído
Confiabilidade	Protocolo
Consultative Committee for International Tele-	Provedor de acesso à Internet (ISP)
graphy and Telephony (CCITT)	Provedor nacional de acesso (NSP)
CSNET	Provedores de acesso local à Internet
Dados	Receptor
Electronic Industries Association (EIA)	Rede
Entidade	Rede de longa distância – Wide Area Network (WAN)
Federal Communications Commission (FCC)	Rede de área local – Local Area Network (LAN)
Fonte	Rede Metropolitana – Metropolitan Area Network (MAN)
Fórum	Request for Comment (RFC)
<i>Hub</i>	Segurança
Imagen	Semântica
Institute of Electrical and Electronics Engineers (IEEE)	Sintaxe
International Organization for Standardization (ISO)	Telecomunicações
International Telecommunication Union – Tele-	Temporização (<i>timing</i>)
communication Standards Sector (ITU-T)	Topologia em anel
Internet	Topologia em barramento
Internet <i>draft</i> (minuta)	Topologia em estrela
<i>Internetwork</i> (internet)	Topologia em malha
ISPs regionais	Topologia Física
Meio de transmissão	Transmission Control Protocol/Internetworking Protocol (TCP/IP)
Mensagem	Vídeo
Modo <i>full-duplex</i>	

1.6 RESUMO

- Comunicação de dados é o processo de transferência de dados de um dispositivo a outro através de algum meio de transmissão.
- Um sistema de comunicação de dados deve transmitir dados ao destino correto, de modo preciso e em tempo hábil.
- Os cinco componentes básicos de um sistema de comunicação de dados são a mensagem (informação), a fonte, o destino, o meio e o protocolo.
- Textos, números, imagens, áudio e vídeo – são formas diferentes de informação.
- O fluxo de dados entre dois dispositivos pode acontecer de três modos: *simplex*, *half-duplex* ou *full-duplex*.
- Uma rede é um conjunto de dispositivos de comunicação conectados através de algum tipo de meio (os *links*).
- Numa conexão ponto a ponto dois, e somente dois, dispositivos são conectados através de *links* dedicados. Numa conexão multiponto, três dispositivos ou mais compartilham o mesmo *link*.
- A topologia se refere ao arranjo físico ou lógico de uma rede. Dispositivos podem ser dispostos em rede segundo as topologias em malha, estrela, barramento ou anel.
- Uma rede pode ser classificada como rede local (Local Area Network – LAN), rede metropolitana (Metropolitan Area Network – MAN) ou rede de longa distância (Wide Area Network – WAN).

- Uma LAN é um sistema de comunicação de dados abrangendo um edifício, uma planta, um campus ou construções adjacentes.
- Uma MAN é um sistema de comunicação de dados cobrindo uma área do tamanho uma cidade.
- Uma WAN é um sistema de comunicação de dados que interliga estados, países ou todo o planeta.
- Uma internet é uma rede de redes.
- A Internet é uma coleção de muitas redes separadas.
- TCP/IP é o protocolo de acesso à Internet.
- Existem provedores local, regional, nacional e internacional de acesso à Internet (ISPs).
- Um protocolo é um conjunto de regras que governa a comunicação de dados; os elementos chave de um protocolo são a sintaxe, a semântica e a temporização (*timing*).
- Os padrões são necessários pois garantem que produtos de diferentes fabricantes coexistam.
- A ISO, ITU-T ANSI, IEEE e EIA são algumas das organizações envolvidas na criação de padrões.
- Fóruns são grupos de interesse pessoal que avaliam e padronizam rapidamente uma nova tecnologia.
- Um Request for Comment é uma idéia ou conceito lançado a padrão da Internet por um precursor.

1.7 PRATIQUE OS CONHECIMENTOS ADQUIRIDOS

Questões de Revisão

1. Identifique os cinco componentes de um sistema de comunicação de dados.
2. Quais são as vantagens de um sistema de processamento distribuído?
3. Quais os três critérios que medem a eficiência de uma rede?
4. Quais são as vantagens de conexão multiponto em relação à conexão ponto a ponto?
5. Quais são os dois tipos possíveis de conexão?
6. Classifique as quatro topologias básicas em termos dos tipos de conexão.
7. Qual é a diferença entre o modo de transmissão *half-duplex* e *full-duplex*?
8. Cite o nome e uma vantagem de cada uma das quatro topologias básicas de rede.
9. Para n dispositivos ligados em rede, qual é o número de cabos necessários para conectar esses dispositivos formando topologias em malha, estrela, barramento e anel?
10. Quais são os fatores determinantes para um sistema de comunicação de dados ser considerado LAN, MAN ou WAN?
11. O que é uma internet? O que é a Internet?
12. Por que as redes precisam de protocolos?
13. Por que as redes precisam ser padronizadas?

Questões de Múltipla Escolha

14. Um _____ é o caminho físico onde viaja a informação.
 - Protocolo
 - Meio
 - Sinal
 - Todos acima
15. A informação a ser trocada num sistema de comunicação de dados recebe o nome de _____.
 - Meio
 - Protocolo
 - Mensagem
 - Transmissão
16. Frequência de falhas e tempo de recuperação da rede pós falha são medidas da _____ de uma rede.
 - Performance
 - Confiabilidade
 - Segurança
 - Todos acima
17. Um usuário não autorizado é uma questão de _____ da segurança.
 - Performance
 - Confiabilidade

- c. Segurança
d. Todos acima
18. Que topologia requer um concentrador ou *hub*?
a. Malha
b. Estrela
c. Barramento
d. Anel
19. Que topologia requer uma conexão multiponto?
a. Malha
b. Estrela
c. Barramento
d. Anel
20. A comunicação entre um computador e um teclado envolve uma transmissão _____.
a. *Simplex*
b. *Half-duplex*
c. *Full-duplex*
d. Automática
21. Numa rede com 25 computadores, que topologia requer a maior quantidade de cabos para ser implementada?
a. Malha
b. Estrela
c. Barramento
d. Anel
22. Uma transmissão de TV é um exemplo de transmissão _____.
a. *Simplex*
b. *Half-duplex*
c. *Full-duplex*
d. Automática
23. Uma conexão _____ provê um *link* dedicado entre dois dispositivos.
a. Ponto a ponto
b. Multiponto
c. Primária
d. Secundária
24. Numa conexão _____ dois dispositivos ou mais podem compartilhar o mesmo *link*.
a. Ponto a ponto
b. Multiponto
c. Primária
d. Secundária
25. Numa transmissão _____, toda a capacidade do canal é compartilhada, durante todo o tempo, pelos dois dispositivos que estiverem comunicando entre si.
a. *Simplex*
b. *Half-duplex*
c. *Full-duplex*
d. *Half-simplex*
26. Um rompimento de cabo numa topologia em _____ pára toda a comunicação.
a. Malha
b. Barramento
c. Estrela
d. Primária
27. Que organização tem autoridade sobre o comércio nacional e internacional no campo das comunicações?
a. ITU-T
b. IEEE
c. FCC
d. ISO

Exercícios

28. Imagine seis computadores conectados numa topologia em malha. Quantos cabos são necessários? Quantas interfaces de rede são necessárias para cada computador?
29. Se uma conexão apresentar falhas, discuta as possíveis consequências para cada uma das quatro redes a seguir.
a. Cinco dispositivos organizados numa topologia em malha.
b. Cinco dispositivos organizados numa topologia em estrela (sem conectar o *hub*).
- c. Cinco dispositivos organizados numa topologia em barramento.
d. Cinco dispositivos organizados numa topologia em anel.
30. Desenhe uma topologia híbrida com um *backbone* em estrela e três redes em anel.
31. Desenhe uma topologia híbrida com um *backbone* em anel e duas redes em barramento.
32. Desenhe uma topologia híbrida com um *backbone* em barramento conectando dois anéis *backbones* em anel.

- onde cada *backbone* em anel conecta três redes em estrela.
- 33. Desenhe uma topologia híbrida com um *backbone* em estrela conectando dois *backbones* em barramento, onde cada *backbone* em barramento conecta três redes em anel.
 - 34. Encontre três padrões definidos pela ISO.
 - 35. Encontre três padrões definidos pela ITU-T.
 - 36. Encontre três padrões definidos pela ANSI.
 - 37. Encontre três padrões definidos pela IEEE.
 - 38. Encontre três padrões definidos pela EIA.
 - 39. Cite dois exemplos de como as redes fazem parte de sua vida hoje.
 - 40. Quando uma primeira pessoa faz uma chamada telefônica local para uma segunda pessoa está sendo estabelecida uma conexão ponto a ponto ou multiponto? Explique sua resposta.

Arquiteturas de Redes

Uma rede de computadores utiliza a combinação *hardware + software* para enviar dados de um local a outro. É considerado *hardware* da rede toda a infra-estrutura física (os equipamentos) para transportar sinais de um local a outro. Além do *hardware*, precisamos do *software* para viabilizar o processo de comunicação, porque são esperados serviços numa rede muito mais complexos do que simplesmente enviar um sinal de um computador fonte para um computador destino.

Podemos comparar a tarefa de implementar a conectividade de uma rede com a solução matemática de um problema do ponto de vista computacional. Fundamentalmente, todo o problema matemático é resolvido num computador através dos recursos de *hardware*. Entretanto, esse tipo de trabalho pode ser um tanto tedioso se dispusermos apenas do *hardware*. É necessário conhecer todos os recursos físicos disponíveis no computador, como memória de armazenamento de dados e recursos para manipulação de dados nativos do processador, além de todos os dispositivos de entrada/saída (E/S) que forem utilizados, para resolver um problema partindo apenas do referencial do *hardware*. A tarefa torna-se mais simples se um *software* conveniente é introduzido nesse cenário. Um programa de computador, escrito numa linguagem de alto nível, pode resolver o problema diretamente, sem o conhecimento específico do *hardware*. Nesse sentido, os detalhes de como a solução é levada a cabo pelo *hardware* podem ser deixados para as camadas ou níveis mais elevados do *software* empregado.

O problema anterior assemelha-se ao da conectividade numa rede de computadores. A tarefa simples de enviar um *e-mail* para um outro país do mundo pode ser dividida em muitas sub-tarefas, cada qual realizada por um pacote de *software* diferente. Cada *software* se utiliza dos serviços de outro pacote de *software* para realizar a tarefa que lhe é confiada. Na camada mais baixa, o sinal ou conjunto de sinais referentes ao processo de troca de *e-mails* é enviado do computador fonte ao computador destino.

Neste capítulo, daremos uma idéia geral das camadas de uma rede e discutiremos as funcionalidades de cada uma. Descrições detalhadas dessas camadas são deixadas para os próximos capítulos.

2.1 MODELO DE CAMADAS

Utilizamos os conceitos de camadas diariamente em nossas vidas. Vamos, por exemplo, considerar duas amigas que freqüentemente utilizam os Correios para trocar correspondências. O processo de enviar uma simples carta a um amigo seria complexo se não dispuséssemos dos serviços dos Correios. A Fig. 2.1 mostra como essa tarefa é realizada.

Emissor, Receptor e Meio de Transporte

Na Fig. 2.1, está clara a existência de um emissor, um receptor e um meio para transportar a carta. Nesse sentido, existe uma hierarquia entre as tarefas a serem executadas.

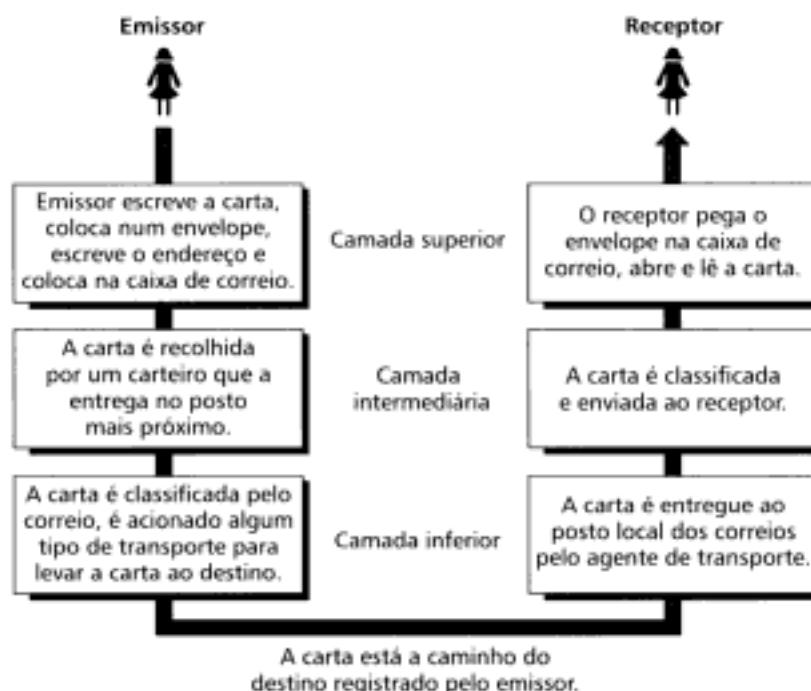


Figura 2.1 Enviando uma carta.

No Lado do Emissor

Vamos primeiramente descrever, em ordem, as tarefas desenvolvidas no lado de quem envia a carta.

- **Camada superior.** O emissor escreve a carta; coloca a carta num envelope; escreve os endereços do remetente e do destinatário e deposita a carta na caixa de correio.
- **Camada intermediária.** A carta é recolhida por um carteiro ou um agente responsável dos Correios que a entrega a um posto mais próximo.
- **Camada inferior.** A carta é classificada pelo Correio; nesse ponto é acionado algum tipo de transporte para levar a carta ao destino.

Ao Longo do Caminho

A carta está a caminho do destino registrado pelo emissor. Dependendo do caso, a carta pode ser repassada ao destino pelo próprio posto dos Correios que a recebeu ou pode ser reenviada a um posto central, para que a encomenda seja entregue ao destinatário. Além disso, a carta pode ser transportada através de caminhão, trem, avião, navio ou uma combinação desses meios de transporte.

No Lado do Receptor

- **Camada inferior.** A carta é entregue ao posto local dos Correios pelo agente que a transportou.
- **Camada intermediária.** A carta é classificada e enviada para a caixa de correio do receptor.
- **Camada superior.** O receptor pega o envelope na caixa de correio, abre o envelope e lê a carta.

Hierarquia

De acordo com nossa análise, há três atividades diferentes no lado emissor, assim como no lado receptor. A tarefa de transportar a carta entre o emissor e o receptor é desempenhada pelo carteiro ou agente responsável nos Correios. Na análise, parece óbvio que as tarefas de envio e recebimento da carta obedecem uma ordem ou hierarquia. No lado emissor, a carta deve ser escrita e colocada na caixa de correio antes do carteiro ou agente dos Correios responsável pegá-la e entregá-la ao posto de recebimento mais próximo. No lado receptor, a carta deve ter sido deixada na caixa de correio do receptor antes de ser retirada e lida pelo destinatário.

Serviços

Cada camada do lado do emissor utiliza os serviços da camada imediatamente abaixo dela. O emissor, no referencial da camada superior, usa os serviços da camada intermediária. A camada intermediária usa os serviços da camada inferior. A camada inferior usa os serviços de um carteiro ou agente responsável pelo transporte da carta. No lado receptor, a análise deve ser realizada de baixo para cima, isto é, da camada inferior para a camada superior, até que o destinatário leia as informações escritas na carta.

2.2 MODELO DE CAMADAS DA INTERNET

Existe um modelo de cinco camadas* que domina os processos de comunicação de dados e a conectividade entre um emissor e um receptor numa *internetworking*: o **Modelo da Internet** ou **pilha de protocolos TCP/IP** (veja Fig. 2.2). Esse modelo é composto de cinco camadas devidamente ordenadas: física (camada 1), enlace de dados (camada 2), rede (camada 3), transporte (camada 4) e aplicação (camada 5). A Figura 2.3 ilustra o relacionamento entre as camadas quando uma mensagem é enviada de um dispositivo A a outro dispositivo B. Quando a mensagem viaja de A até B, ela pode passar através de muitos nós intermediários. Estes nós intermediários possuem somente as três primeiras camadas do modelo TCP/IP.

No desenvolvimento do modelo, os projetistas fragmentaram o processo de transmissão de dados num conjunto fundamental de elementos. Eles identificaram quais funções da rede possuíam algum correlacionamento e agruparam essas funções em grupos discretos que se tornaram as camadas do modelo. Assim, cada camada adquiriu uma família de funcionalidades distintas das demais. Então, organizando adequadamente as diversas funcionalidades em cada camada, um comitê organizou a arquitetura do modelo da Internet e deu-lhe flexibilidade suficiente ao ponto de ganhar a notoriedade que vemos hoje.

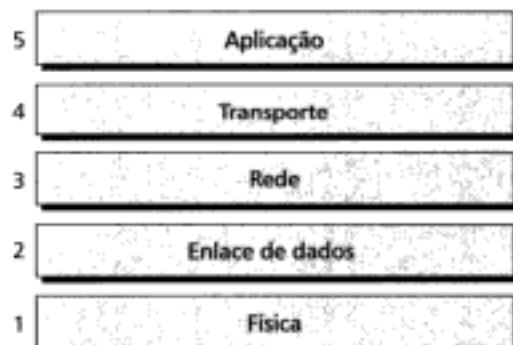


Figura 2.2 Modelos de camadas da Internet.

* N. de R. T.: Muitos preferem usar o termo nível no lugar do termo camada.

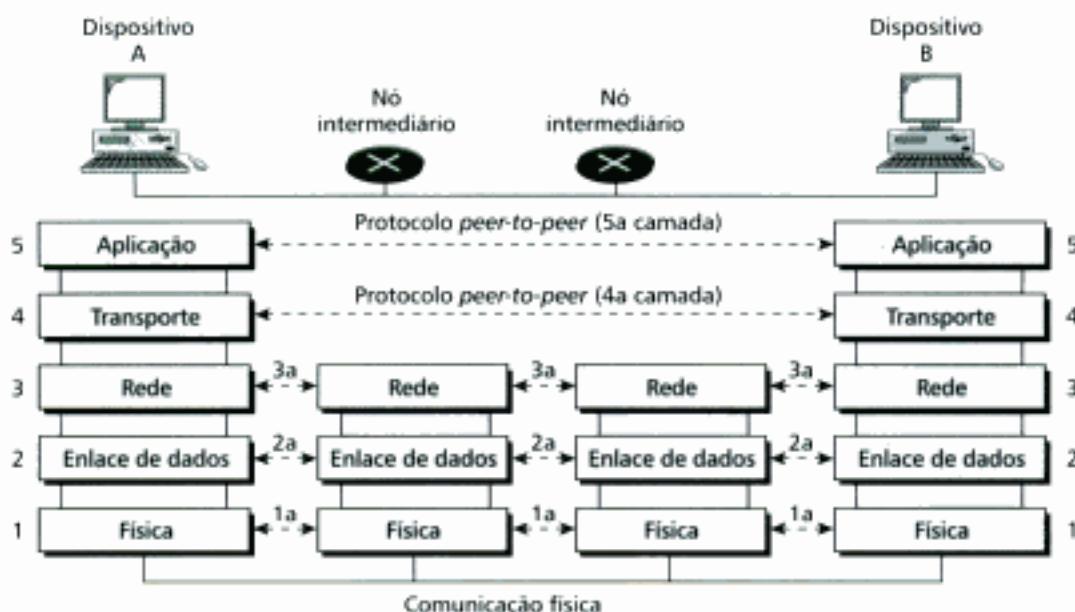


Figura 2.3 Processos peer-to-peer.

Dentro de uma única máquina, os níveis mais altos do modelo sempre chamam os serviços dos níveis mais baixos. Por exemplo, a camada 3 usa os serviços disponíveis na camada 2 e provê serviços para a camada 4. Entre duas ou mais máquinas, a camada x em uma máquina sempre se comunica com a camada x da máquina para onde seguem os dados. Esta comunicação é controlada por regras e convenções denominadas protocolos. Os processos em cada máquina que se comunicam numa mesma camada são denominados **processos peer-to-peer**. A comunicação entre máquinas forma então um grande processo peer-to-peer, usando os protocolos apropriados em cada camada.

Processos Peer-to-Peer

Na camada física, a comunicação acontece diretamente: na Figura 2.3, o dispositivo A envia uma cadeia de *bits* (um pacote) ao dispositivo B. Contudo, nas camadas mais altas, a comunicação deve acontecer entre camadas, de cima para baixo no dispositivo A e na ordem inversa no dispositivo B. Cada camada no lado do dispositivo transmissor (dispositivo A) adiciona sua própria informação à mensagem recebida da camada logo acima e transfere todo o pacote para a camada imediatamente abaixo.

Na camada 1, o pacote como um todo é convertido num sinal elétrico a ser transmitido para o dispositivo receptor. Na máquina receptora (dispositivo B), a mensagem é desempacotada (aberta) camada por camada, onde cada processo recebe e remove apenas os dados destinados a ele. Por exemplo, a camada 2 remove os dados destinados a ela e passa o restante do pacote à camada 3 que, por sua vez, remove os dados destinados a ela e passa o restante do pacote à camada 4 e assim por diante até a última camada.

Interfaces entre Camadas

A passagem de dados e informação de rede através das camadas do dispositivo transmissor (A) e a respectiva recuperação da informação nas camadas do dispositivo receptor (B) somente é possível graças a uma **interface** entre cada par de camadas adjacentes. Cada interface define que tipo de informação e serviços uma camada deve proporcionar à camada imediatamente acima. Interfaces bem definidas, aliadas às funcionalidades das camadas, dão modularidade a uma rede. Uma vez que a função de uma camada é fornecer os serviços esperados pela camada imediatamente acima, a função específica de uma determinada camada pode ser modificada ou substituída sem que hajam mudanças nas demais camadas da vizinhança.

Organização das Camadas

As cinco camadas podem ser imaginadas como parte de três subgrupos. As camadas 1, 2 e 3 – física, enlace de dados e de rede, respectivamente – são as camadas de suporte à rede. Elas lidam com os aspectos físicos da movimentação de dados de um dispositivo a outro (tais como especificações elétricas, conexões físicas, endereçamento físico e lógico, sincronização do transporte e confiabilidade). A camada 5 – aplicação – pode ser tratada como camada de suporte ao usuário. Ela permite a interoperabilidade entre sistemas incompatíveis do ponto de vista de *software*. A camada 4 – transporte – é o elo entre os dois subgrupos anteriores. Ela verifica qual das camadas inferiores fez o chamado da transmissão e que tem os dados numa forma que as camadas superiores possam utilizar.

A Figura 2.4 dá uma visão ampla das camadas do modelo TCP/IP. O pacote de dados em L5 representa o encapsulamento dos dados na camada 5, o pacote de dados em L4 representa o encapsulamento dos dados na camada 4 e assim por diante. O processo de comunicação origina-se na camada de aplicação (camada 5) e então move-se para baixo seqüencialmente, de camada em camada. Em cada camada pode ser adicionado um **cabeçalho (header)** ao pacote encapsulado. Na camada 2, pode ainda ser adicionado um **campo de detecção de erros***. Nesse nível, os dados encapsulados passam a ser denominados quadro ou *frame* e estão prontos para serem convertidos em sinais eletromagnéticos e transportados pelo meio físico de comunicação.

Após alcançar o destino, o sinal eletromagnético entra pela camada 1 e é reconvertido para a forma digital (em quadros ou *frames*). Desse modo, os dados estão prontos para serem movidos novamente através das camadas no lado do receptor. Tão logo o pacote de dados alcance a camada de enlace do receptor, os cabeçalhos e campos adicionados nesse mesmo nível no lado do transmissor são removidos e as decisões apropriadas são tomadas, baseadas na informação contida nos cabeçalhos. Chegando à camada 5, a mensagem está novamente numa forma apropriada para a aplicação que fará uso dela e a disponibilizará ao receptor.

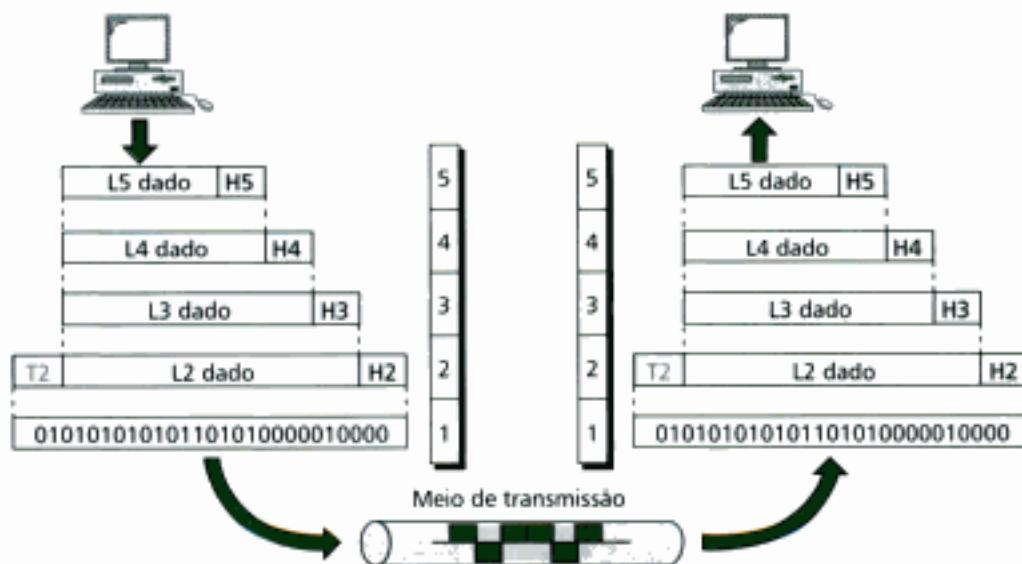


Figura 2.4 Transmissão usando modelo da Internet.

Funções das Camadas

Nessa seção, faremos uma breve discussão das funções de cada camada.

* N. de R. T.: Este campo é denominado FCS (Frame Check Sequence) e a função dele é controlar/detectar erros nos quadros de dados manipulados na camada de enlace.

Camada Física

A **camada física** coordena as funções exigidas para transmitir uma cadeia de *bits* num meio físico específico. Nela são tratadas as especificações elétricas e mecânicas de uma interface e do meio de transmissão. Ela também define os procedimentos e funcionalidades que os dispositivos físicos e interfaces devem possuir para tornar possível a comunicação. A Figura 2.5 mostra a posição da camada física com respeito aos meios de transmissão e a camada de enlace de dados.

Discutiremos a camada física detalhadamente na Parte II deste livro, juntamente com os protocolos predominantes nessa camada. Entretanto, as funções da camada física podem ser sintetizadas da seguinte forma:

- **Características físicas das interfaces e dos meios.** A camada física define as características mecânicas e elétricas da interface entre o dispositivo que transmite e os meios de transmissão. Ela também define que tipo de meio de transmissão deve ser utilizado (veja Capítulo 7).
- **Representação dos dados.** Os dados na camada física estão dispostos numa cadeia de *bits* (seqüência de 0s e 1s) sem qualquer interpretação. Para serem transmitidos, os *bits* devem ser codificados em sinais – elétricos ou ópticos. A camada física define o tipo de representação dos dados (como os 0s e 1s são convertidos em sinais elétricos ou ópticos).
- **Taxa de transferência de dados.** A **taxa de transmissão** – o número de *bits* enviados por segundo – também é definida na camada física. Em outras palavras, a camada física define o tempo de duração de um *bit* no meio.
- **Sincronização dos bits.** O transmissor e o receptor não devem somente usar a mesma taxa de transmissão, mas devem estar sincronizados no nível dos *bits*. Em outras palavras, os relógios (*clocks*) do transmissor e do receptor devem estar sincronizados.

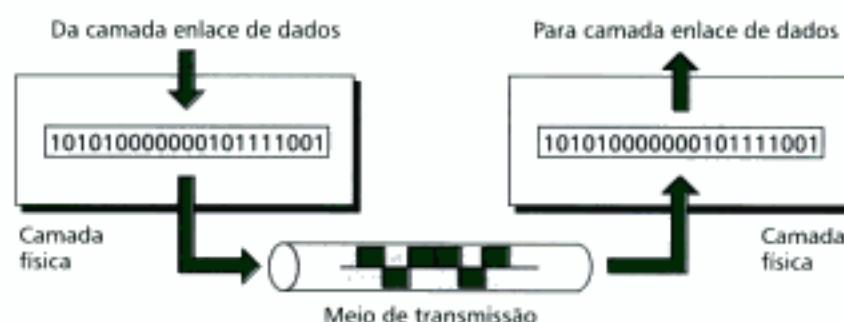


Figura 2.5 Camada física.

A camada física é responsável pela transmissão individual dos *bits* de um nó a outro numa rede.

Camada de Enlace de Dados

A **camada de enlace de dados** converte os dados brutos e não confiáveis oriundos da camada física, num link confiável para a camada imediatamente acima (a camada de rede). Assim, ela assegura que os dados da camada física cheguem livres de erros à camada de rede. A Figura 2.6 mostra o relacionamento da camada de enlace de dados com as camadas de rede e física.

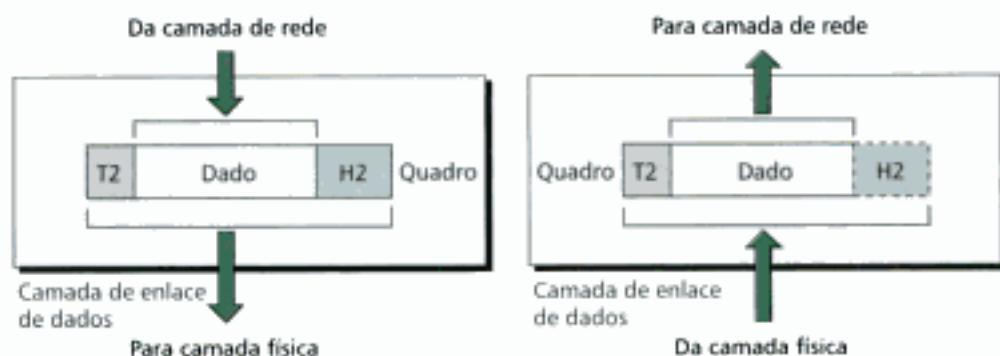


Figura 2.6 Camada de enlace de dados.

A camada de enlace de dados é responsável pela transmissão de quadros entre os nós de uma rede.

Discutiremos a camada de enlace detalhadamente na Parte III deste livro, juntamente com os protocolos predominantes nessa camada. Entretanto, as funções da camada de enlace podem ser sintetizadas da seguinte forma:

- **Enquadramento (*framing*)**. A camada de enlace de dados divide a cadeia de *bits* recebidos da camada de rede em unidades de dados gerenciáveis denominados **quadros** ou *frames*.
- **Endereçamento físico**. Se os quadros tiverem que ser distribuídos para diferentes sistemas na rede, a camada de enlace adiciona um cabeçalho a cada quadro para definir o transmissor e/ou o receptor de quadro específico. Se a intenção é enviar um quadro para uma rede fora do domínio do transmissor, o endereço do receptor é o endereço do dispositivo que conecta as duas redes, ou seja, o endereço de um dispositivo intermediário que interliga as redes do transmissor e do receptor.
- **Controle de fluxo**. Se a taxa de transmissão de dados no transmissor for maior que a taxa de recepção dos dados no receptor, a camada de enlace utiliza um mecanismo de controle para controlar o fluxo de dados e prevenir sobrecarga de dados no receptor.
- **Controle de erro**. A camada de enlace adiciona confiabilidade aos dados recebidos da camada física através de um mecanismo de detecção, perdas e de retransmissão de quadros. Ela também se utiliza de um mecanismo para evitar duplicação de quadros. O controle de erro normalmente é adicionado num campo no final do quadro.
- **Controle de acesso**. Quando dois ou mais dispositivos estão conectados ao mesmo *link*, os protocolos da camada de enlace determinam qual dispositivo mantém o controle sobre o *link* num dado instante de tempo.

A Figura 2.7 ilustra um processo tipo ***hop-to-hop (node-to-node)***.

Exemplo 1

Na Figura 2.8, o nó de endereço físico 10 envia um quadro de dados para outro nó cujo endereço físico é 87. Os dois nós estão conectados através do mesmo *link*. No nível de enlace, este quadro possui o endereço físico localizado no cabeçalho. Estes são os únicos endereços necessários para que a comunicação entre os dois seja possível. O resto do cabeçalho contém outras informações que devem ser tratadas no nível de enlace. O campo no final do quadro contém usualmente informações necessárias à detecção de erros.

Camada de Rede

A **camada de rede** assegura o roteamento dos pacotes **da fonte ao destino**, possivelmente através de inúmeras redes. Considerando que a camada de enlace supervisiona a entrega de pacotes

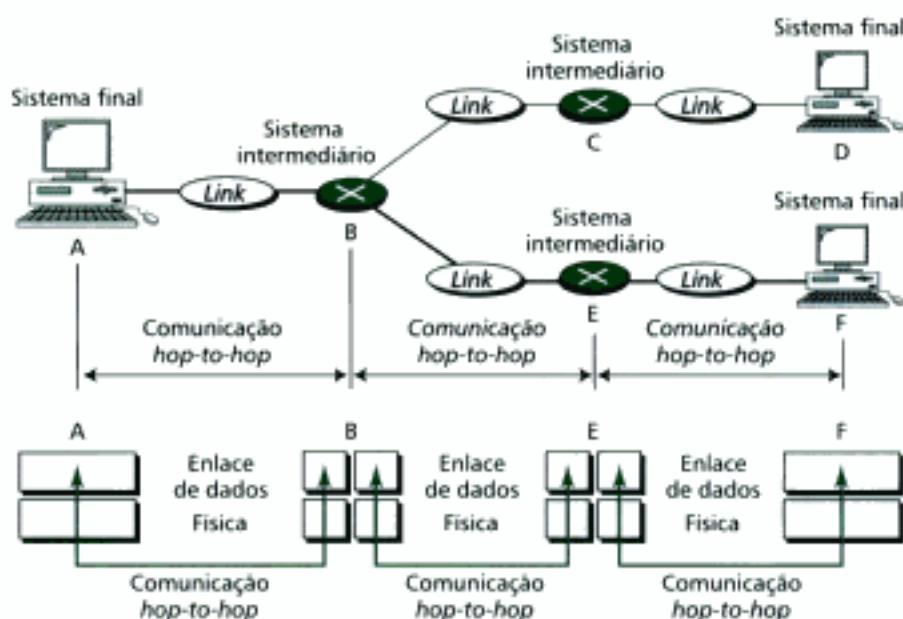


Figura 2.7 Comunicação hop-to-hop (node-to-node).

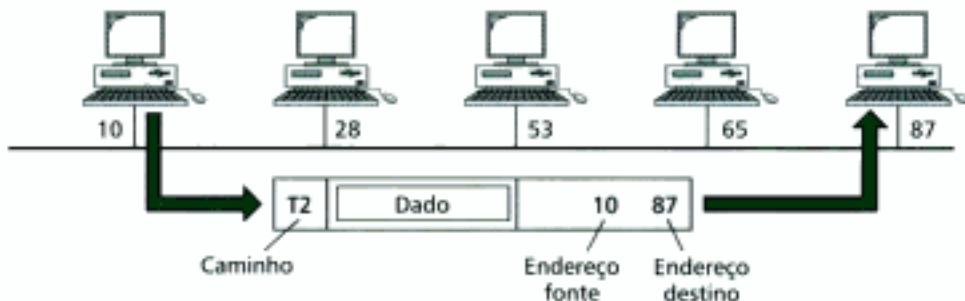


Figura 2.8 Exemplo 1.

entre dois sistemas diferentes na mesma rede, a camada de rede assegura que cada pacote consiga sair do dispositivo de origem e chegar ao dispositivo ou destino final.

- Se dois sistemas estiverem conectados no mesmo segmento de rede, usualmente não haverá necessidade da camada de rede. Contudo, se dois sistemas estiverem conectados em redes diferentes com algum dispositivo ativo de rede interligando-os, a camada de rede será fundamental para que o pacote saia da rede de origem e chegue à rede de destino. A Figura 2.9 ilustra o relacionamento entre a camada de rede (3) e as camadas de enlace (2) e de transporte (4).

Discutiremos a camada de rede detalhadamente na Parte IV deste livro, juntamente com os protocolos predominantes nessa camada. Entretanto, as funções da camada de rede podem ser sintetizadas da seguinte forma:

- **Endereçamento lógico.** O endereçamento físico implementado na camada de enlace de dados resolve localmente o problema de endereçamento na rede. Se um pacote tiver que deixar o ambiente local da rede é necessário outro mecanismo de endereçamento para fazer distinção entre a fonte local e o destino remoto dos dados. A camada de rede adiciona um cabeçalho ao pacote que chega da camada de transporte incluindo, dentre outras coisas, o endereço lógico do dispositivo que envia e do dispositivo que recebe o pacote de dados.

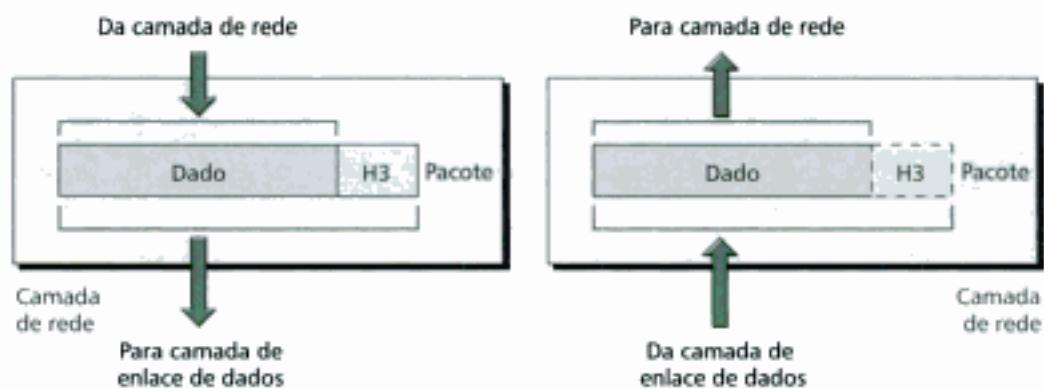


Figura 2.9 Camada de rede.

A camada de rede é responsável pelo roteamento dos pacotes na *internetworking*.

- **Roteamento (routing).** Quando interligamos redes ou *links* diferentes para criar uma *internetworking* (uma rede de redes), os dispositivos inter-redes ou ativos de rede, os *routers* e os *switches* de camada 3, roteiam ou comutam os pacotes até o destino final. Uma das funções principais da camada de rede é proporcionar esse mecanismo de roteamento de pacotes.

A Figura 2.10 detalha um processo de entrega de pacotes através de uma *internetworking*.

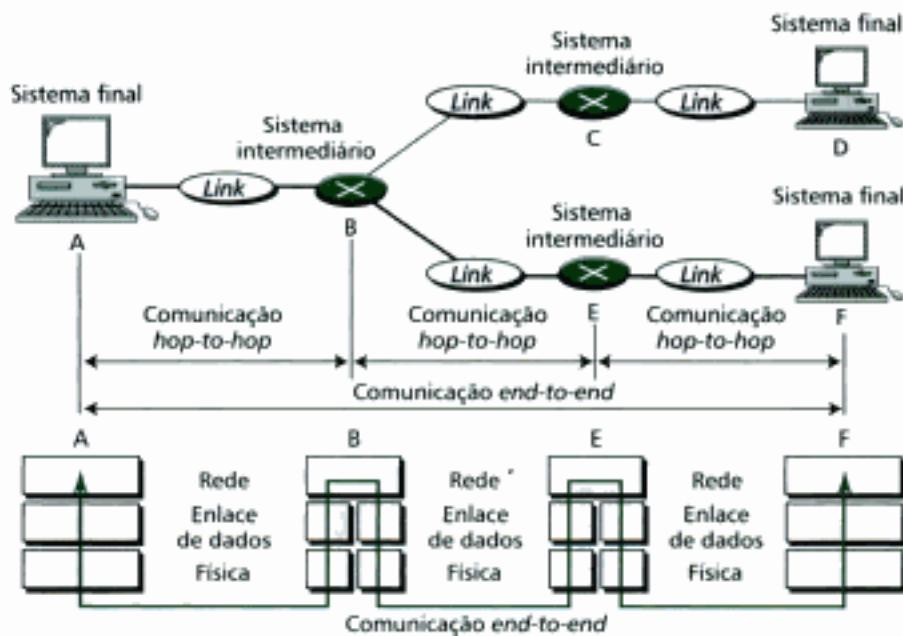


Figura 2.10 Comunicação origem-destino.

Exemplo 2

Na Figura 2.11, desejamos enviar dados a partir do nó cujos endereços físico e de rede são 10 e A, respectivamente, localizados numa LAN, para um nó cujos endereços físico e de rede são 95 e P, respectivamente, localizados noutra LAN. Dada a localização em redes diferentes, o endereço físico não é suficiente para garantir a entrega dos pacotes de dados, já que a entrega através do endereço físico fica restrita internamente em

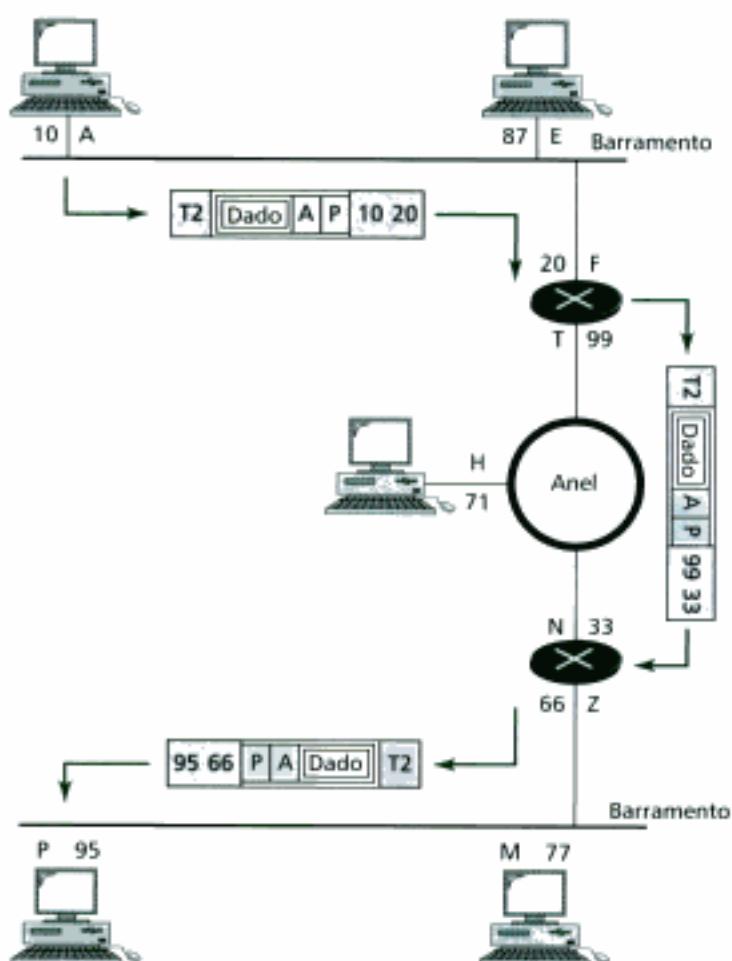


Figura 2.11 Exemplo 2.

cada LAN. Desse modo, é necessário um endereço universal, válido em qualquer domínio fora da rede local, para entrega dos pacotes. O endereço lógico possui universalidade. O pacote parte da camada de rede do dispositivo fonte contendo o endereço lógico da origem e do destino (na figura, A e P respectivamente), e mantém esses endereços inalterados quando o pacote atravessa de uma rede para a outra. É claro que, durante os saltos inter-redes, os endereços físicos contidos nos pacotes são alterados. O dispositivo assinalado com um X é um roteador (dispositivo de *internetworking* ou inter-redes) e será discutido no Capítulo 16.

Camada de Transporte

A **camada de transporte** garante a entrega de toda uma mensagem entre **processos finais** (usuários). Considerando que a camada de rede roteia os pacotes individuais da origem ao destino, ela não vê nenhum relacionamento entre os pacotes endereçados logicamente lá. No nível de rede cada pacote é tratado individualmente, como se cada um fizesse parte de mensagens diferentes. A camada de transporte cuida, dentre outras coisas, de assegurar que toda a mensagem chegue intacta e livre de erros, ou seja, controlando erros e fluxo no nível de processos finais. A Figura 2.12 mostra o relacionamento entre a camada de transporte (4) e as camadas de rede (3) e de aplicação (5).

Discutiremos a camada de transporte detalhadamente na Parte V deste livro, juntamente com os protocolos predominantes nessa camada. Entretanto, as funções da camada de transporte podem ser sintetizadas da seguinte forma:

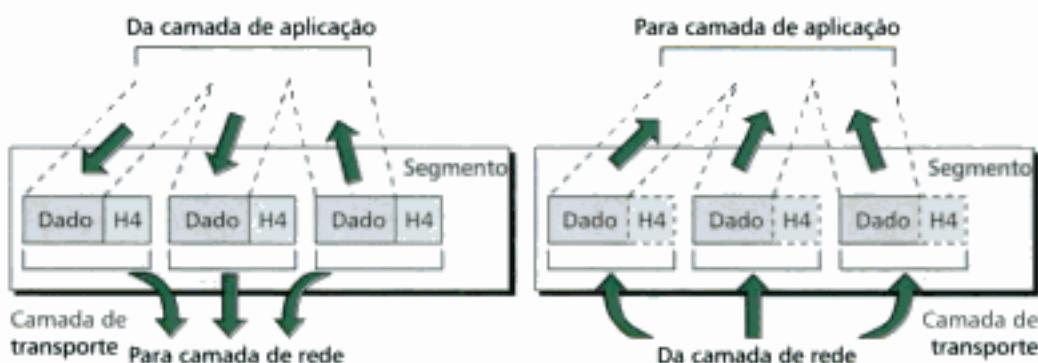


Figura 2.12 Camada de transporte.

A camada de transporte é responsável pela entrega de uma mensagem entre processos finais.

- **Endereçamento de portas.** Computadores freqüentemente rodam muitos processos (programas) ao mesmo tempo. As entregas envolvendo processos finais não se resumem simplesmente ao transporte de dados de um computador a outro, mas envolvem processos específicos em cada um dos computadores onde os processos estão sendo rodados. Desse modo, um cabeçalho na camada de transporte deve incluir um tipo de endereçamento específico denominado **endereço de porta***. A camada de rede encaminha cada pacote para o computador correto; a camada de transporte encaminha toda uma mensagem para o processo correto noutro computador.
- **Segmentação e reagrupamento de pacotes.** Uma mensagem não pode monopolizar o *link* ou segmento de rede por onde trafega. Isso diminui a *performance* da rede. Pensando desse modo, normalmente uma mensagem é dividida em vários segmentos de tamanhos variáveis, onde cada segmento contém um número de identificação. Tais números habilitam a camada de transporte do dispositivo receptor a remontar corretamente a mensagem original e, ainda, identificar e/ou substituir pacotes extraviados durante a transmissão.
- **Controle do link.** A camada de transporte pode ser orientada à conexão ou sem conexão. Um transporte sem conexão trata cada segmento como um pacote independente e os entrega à camada de transporte da máquina de destino. Um transporte orientado à conexão estabelece uma conexão com a camada de transporte da máquina de destino antes de iniciar a entrega dos pacotes. Após o término da transferência de dados a conexão é finalizada.
- **Controle de fluxo.** Assim como na camada de enlace, a camada de transporte também faz controle de fluxo. Entretanto, o controle de fluxo nessa camada é realizado fim a fim ao invés de ser através de um único *link*.
- **Controle de erros.** Como na camada de enlace, a camada de transporte faz controle de erros. Contudo, o controle de erro nessa camada é realizado fim a fim ao invés de ser através do *link*. A camada de transporte do dispositivo de origem assegura que toda a mensagem chegue ao destino (a camada de transporte do dispositivo destino) livre de erros (dano, perda ou duplicação). A correção de um erro normalmente se faz através de um pedido de retransmissão do segmento.

A Figura 2.13 ilustra uma entrega entre processos finais das camadas de transporte.

* N. de R. T.: Também denominado número de porta.



Figura 2.13 Segurança no processo entre processos finais na comunicação de mensagem.

Exemplo 3

A Figura 2.14 mostra um exemplo de comunicação entre as camadas de transporte. Os dados provenientes das camadas superiores do dispositivo transmissor (A) recebem um cabeçalho identificando os endereços das portas j e k (onde j é o endereço do processo que envia os dados e k é o endereço do processo que deve recebê-los). Visto que o tamanho total do pacote dados a ser transmitido é maior que o tamanho de encapsulamento permitido para a camada de rede, os dados são divididos em dois pacotes, cada qual retendo os endereços originais das portas j e k . Na camada de rede são adicionados a cada um dos pacotes os endereços lógicos de origem e destino (A e P). Assim, os pacotes podem ser entregues por diferentes rotas na rede e, até mesmo, chegar ao destino na ordem de sequência original ou fora dela. Os dois pacotes são entregues à camada de transporte do dispositivo receptor (P) que é responsável por remover os cabeçalhos contendo os endereços das portas, reagrupar os dois pacotes para recompor a mensagem e entregar a mensagem ao processo apropriado na camada de aplicação.

Camada de Aplicação

A **camada de aplicação** permite ao usuário final o acesso à rede (seja ele humano ou outro *software*). Ela provê interfaces e suporta serviços, tais como *e-mail*, acesso e transferência de arquivos, *log-in* remoto, acesso à World Wide Web e assim por diante.

A Figura 2.15 mostra o relacionamento da camada de aplicação com os usuários e a camada de transporte.

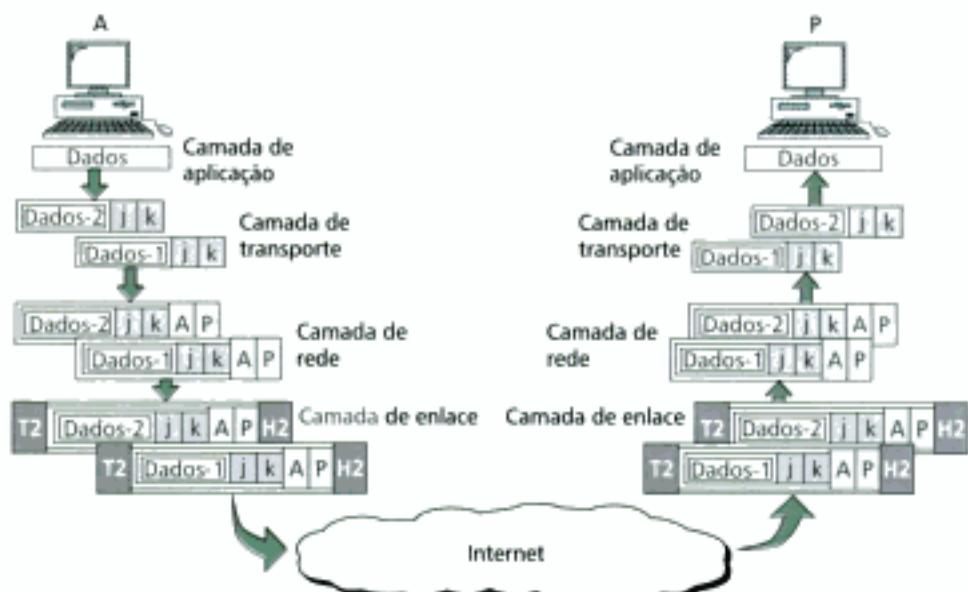


Figura 2.14 Exemplo 3.

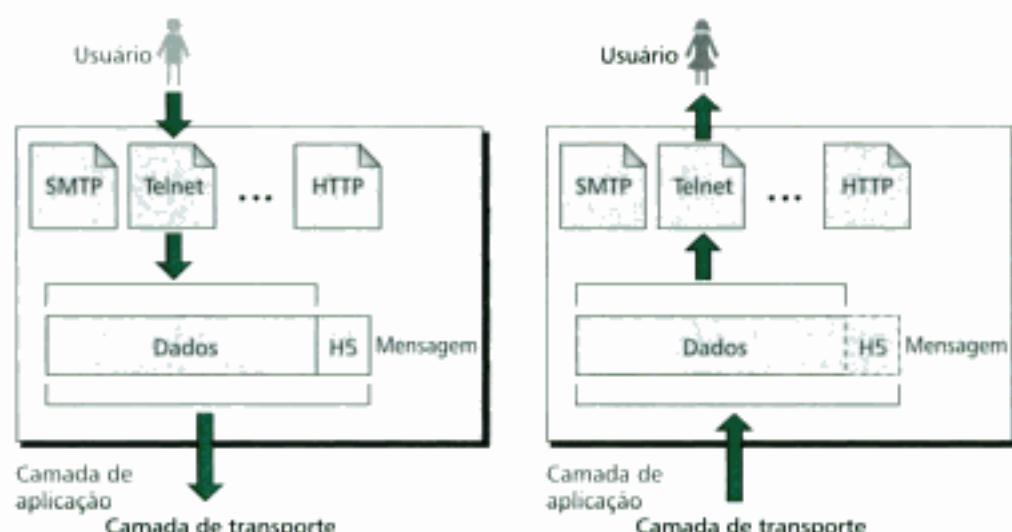


Figura 2.15 Camada de aplicação.

A camada de aplicação é responsável pelos serviços providos ao usuário.

Discutiremos a camada de aplicação em detalhes na Parte VI deste livro, onde também serão incluídos os protocolos predominantes nessa camada. Entretanto, dentre as funções básicas da camada de aplicação podemos citar:

- **Serviços de correio eletrônico (Simple Mail Transfer Protocol – SMTP).** Esta aplicação é a base para troca de *e-mails*.
- **Acesso e transferência de arquivos (File Transfer Protocol – FTP).** Esta aplicação permite ao usuário acessar arquivos em um *host* remoto (lê-los e/ou modificá-los), baixar arquivos de um *host* remoto para usá-los num computador local e gerenciar ou controlar arquivos remotamente.
- **Terminal remoto (Telnet).** Um usuário pode ser autenticado em um computador remoto e acessar os recursos deste computador.
- **Acesso à World Wide Web (HyperText Transfer Protocol – HTTP).** A aplicação mais comum hoje em dia é acessar a World Wide Web (WWW).

Resumo das Camadas

A Figura 2.16 mostra um resumo das funções de cada camada.

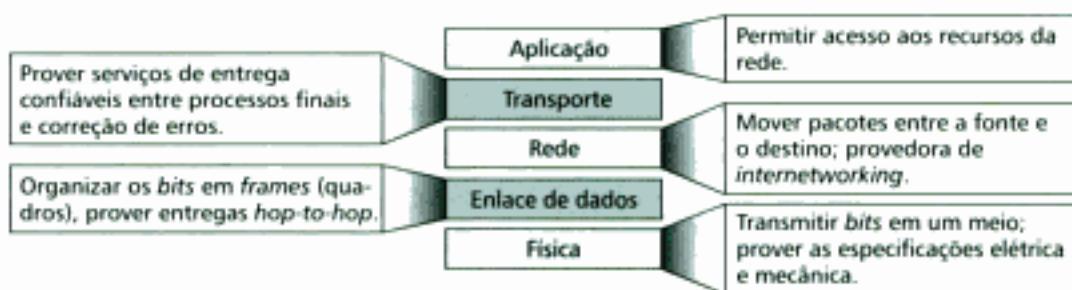


Figura 2.16 Resumo dos serviços das camadas.

2.3 MODELO OSI

O modelo de referência para **Interconexão de Sistemas Abertos (Open Systems Interconnection – OSI)** foi desenvolvido pela International Organization for Standardization (ISO). Trata-se de um modelo de sete camadas. O modelo OSI nunca foi implementado seriamente enquanto pilha de protocolos. Entretanto, serve como modelo teórico ou de referência para os demais e, assim, foi desenvolvido para mostrar como uma pilha de protocolos deveria ser implementada. A Figura 2.17 apresenta as sete camadas do modelo OSI da ISO.

Como pode ser visto na Figura 2.17, o modelo OSI define duas camadas extras: as camadas de sessão e de apresentação. A **camada de sessão** é a controladora de diálogo da rede. Foi desenvolvida para estabelecer, manter e sincronizar a interação entre sistemas de comunicação.

A **camada de apresentação** foi projetada para lidar com a sintaxe e a semântica da informação trocada entre dois sistemas. Foi desenvolvida para conversão entre caracteres (por exemplo, entre ASCII e EBCDIC), criptografia, compressão e descompressão de dados.

O Apêndice C contém uma breve descrição do modelo de referência OSI.

Hoje, entretanto, muitas das funcionalidades destas duas camadas foram incorporadas pelas demais camadas. Por exemplo, o problema da criptografia e da decriptografia* pode ser tratado na camada de aplicação ou transporte. Ainda, dados são comprimidos na camada de aplicação pelos protocolos que agem nesse nível. Por essas razões, concentraremos nosso foco no modelo de cinco camadas da Internet.



Figura 2.17 Modelo OSI.

2.4 TERMOS-CHAVE

Cabeçalho (<i>header</i>)	Controle do <i>link</i>
Camada de aplicação	Endereçamento físico
Camada de apresentação	Endereçamento lógico
Camada de enlace de dados	Endereço ou número de porta
Camada de rede	Entrega entre processos finais
Camada de sessão	Erro
Camada de transporte	<i>Hop-to-hop</i>
Campo (FCS)	<i>Host</i>
Conjunto de protocolos TCP/IP	Interface
Controle de acesso	<i>Internetworking</i>
Controle de erro	Modelo da Internet
Controle de fluxo	Modelo OSI (Open Systems Interconnection)

* N. de R. T.: A forma descriptografia também é aceita.

Node-to-node
Processos *peer-to-peer*
Quadro (*frame*)
Roteamento

Segmentação
Serviço de correio eletrônico

2.5 RESUMO

- Em linhas gerais, o modelo de cinco camadas da Internet provê o desenvolvimento de protocolos de rede inteligíveis.
- As camadas física, de enlace e de rede formam as camadas de suporte à rede.
- A camada de aplicação é a camada de suporte ao usuário.
- A camada de transporte conecta as camadas de suporte à rede com a camada de suporte ao usuário.
- A camada física concentra as funções necessárias para converter e transmitir uma cadeia de *bits* em um meio físico.
- A camada de enlace fornece trânsito de dados confiável através de um *link* físico. Essa camada lida com endereçamento físico, topologia de rede, disciplina de linha, rotificação de erros, entrega ordenada de quadros e controle de fluxo.
- A camada de rede é responsável pelo roteamento de pacotes da origem ao destino através de uma *internetworking*.
- A camada de transporte concentra as funções de entrega de mensagem envolvendo os processos entre usuários finais.
- A camada de aplicação permite que usuários acessem a rede.

2.6 PRATIQUE OS CONHECIMENTOS ADQUIRIDOS

Questões de Revisão

1. Liste as camadas do modelo da Internet.
2. Quais são as camadas de suporte à rede no modelo da Internet?
3. Qual é a camada de suporte ao usuário no modelo da Internet?
4. Qual é a diferença entre as entregas da camada de rede e da camada de transporte?
5. O que é um processo *peer-to-peer*?
6. Como a informação é trocada entre as camadas do modelo da Internet?
7. O que são cabeçalhos e campos? Como eles são adicionados e removidos?
8. Quais são as tarefas dedicadas à camada física no modelo da Internet?

Questões de Múltipla Escolha

15. O modelo da Internet possui _____ camadas.
 - Três
 - Cinco
 - Sete
 - Oito
16. Os serviços de entrega de mensagem entre processos é de responsabilidade da camada _____.
 - de rede
 - de transporte
 - de aplicação
 - física
17. A camada _____ é a camada que interage com o meio de transmissão.
 - de rede
 - de transporte
 - de aplicação
 - física
18. Os serviços de correio eletrônico estão disponíveis para usuários da rede na camada de _____.
 - física
 - enlace
 - transporte
 - aplicação

19. Cabeçalhos são _____ quando pacotes são movidos das camadas mais baixas para as camadas mais altas.
- adicionados
 - removidos
 - rearranjados
 - modificados
20. Cabeçalhos são _____ quando pacotes são movidos das camadas mais altas para as camadas mais baixas.
- adicionados
 - removidos
 - rearranjados
 - modificados
21. A camada de _____ situa-se entre as camadas de rede e de aplicação.
- física
 - enlace
 - transporte
 - nenhuma das respostas anteriores
22. A camada 2 situa-se entre a camada física e a camada de _____.
- rede
 - enlace
 - transporte
 - nenhuma das respostas anteriores
23. Quando transmitimos dados de um dispositivo A para outro dispositivo B, o cabeçalho da camada 4 de A é lido pela camada _____ de B.
- física
 - de transporte
 - de aplicação
 - nenhuma das respostas anteriores
24. A camada _____ converte uma cadeia de *bits* em sinais eletromagnéticos.
- física
 - de enlace
 - aplicação
 - nenhuma das respostas anteriores
25. A camada física concentra as funções de adequar uma cadeia de _____ ao meio físico.
- programas
 - diálogos
 - protocolos
 - bits*
26. Qual camada faz a ligação entre as camadas de suporte à rede e a camada de suporte ao usuário?
- Camada de rede
 - Camada física
 - Camada de transporte
 - Camada de aplicação
27. Qual é a principal função da camada de transporte?
- Serviço de entrega *node-to-node*
 - Serviço de entrega entre processos finais
 - Sincronização
 - Atualiza e mantém tabelas de roteamento
28. Qual, dentre os serviços abaixo, é um serviço da camada de aplicação?
- Terminal remoto
 - Transferência de arquivo
 - Correio eletrônico
 - Todas as respostas anteriores

Exercícios

29. Relacione cada um dos seguintes itens a uma das cinco camadas do modelo da Internet.
- Roteamento de caminho na *internetworking*
 - Controle de fluxo
 - Interface com o mundo físico
 - Provê acesso à rede aos usuários finais
 - Comutação de pacotes
30. Associe cada um dos seguintes itens a uma das cinco camadas do modelo da Internet.
- Transporte confiável de dados entre processos finais
 - Seleção de rede
 - Roteamento
31. Associe cada um dos seguintes itens a uma das cinco camadas do modelo da Internet.
- Comunica diretamente com o programa aplicativo do usuário
 - Correção de erro e retransmissão
 - Interface funcional, mecânica e elétrica
 - Responsável pela entrega de pacotes entre nós adjacentes
 - Reagrupamento de pacotes de dados

PARTE II

CAMADA FÍSICA

Iniciaremos a discussão aprofundada do modelo da Internet a partir da camada mais baixa: a camada física. Esta é a camada que realmente interage com os meios de transmissão, isto é, a parte física que assegura a conectividade entre todos os componentes da rede. Nela, está envolvida com o transporte físico da informação de um nó na rede para o próximo. A Figura 1 contextualiza a camada 1 dentro do modelo da Internet.

A camada física possui tarefas realmente complexas para realizar. A maior delas é prover serviços à camada de enlace de dados. Os dados na camada de enlace formam cadeias binárias de 0s e 1s, organizados em quadros, que devem ser enviados através do meio transmissão. Primeiramente, estas cadeias de 0s e 1s devem ser convertidas em outras entidades: os sinais elétricos ou sinais ópticos. Um dos serviços disponíveis na camada física gera sinais que representam fisicamente cada um quadros montados na camada de enlace.

A camada física também deve cuidar da rede física: o meio de transmissão. O meio de transmissão é uma entidade passiva; não possui nenhum programa ou lógica de controle interno como

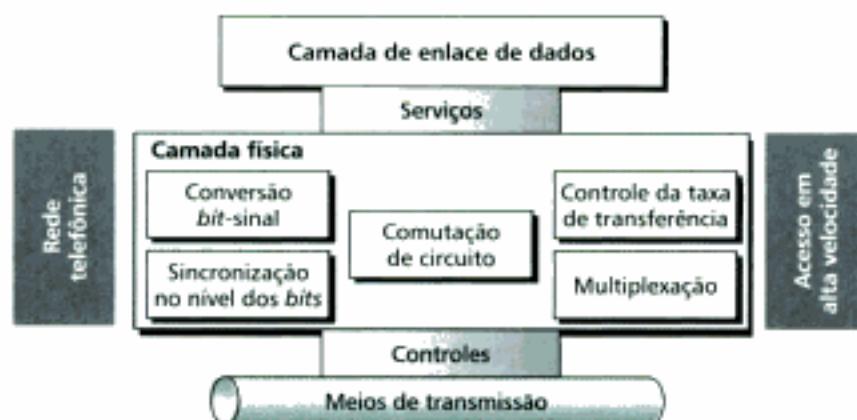


Figura 1 Posição da camada física.

as outras camadas. O meio de transmissão deve ser controlado pela camada física. Além disso, a camada física deve tomar decisão sobre a direção do fluxo de dados. Ainda, nessa camada é que se decide qual o número de canais lógicos para transportar dados oriundos de fontes diferentes.

Serviços

A camada física transfere uma cadeia de *bits* (na forma de um sinal) do transmissor ao receptor. A transferência acontece nó a nó (*node-to-node*) no meio físico. As camadas físicas dos dois nós adjacentes provêem um canal lógico por onde os *bits* podem viajar. A Figura 2 mostra os serviços gerais oferecidos pela camada física.



Figura 2 Serviços.

Conversão Bit-Sinal

O canal lógico abaixo da camada física é o meio de transmissão (cabos ou ar). Como um meio de transmissão (físico) não pode transportar *bits*, precisamos representar os *bits* por um sinal eletromagnético de modo a propagar e a transportar energia através do meio.

Controle da Taxa de Transferência

Embora o meio físico determine o limite superior da taxa de transferência de dados, a camada física é quem tem o controle dessa taxa. Através do projeto de dispositivos da camada física e da implementação de *software* de controle fica determinada a taxa de transferência do meio.

Sincronização no Nível dos Bits

O sincronismo da transferência dos *bits* é crucial na comunicação de dados. A camada física administra a sincronização dos *bits* gerando mecanismos de *clock* que controlam tanto transmissor quanto o receptor.

Multiplexação

A multiplexação é o processo de divisão de um *link* (meio físico) em canais lógicos para melhorar a eficiência da transmissão. A camada física utiliza diversas técnicas para essa finalidade. Embora o meio permaneça o mesmo, o resultado são muitos canais lógicos em vez de um canal físico. As técnicas de multiplexação definidas nessa sessão do livro são necessárias na compreensão dos métodos de acesso nos capítulos futuros.

Comutação

O chaveamento na comunicação de dados pode ser feito em diversas camadas diferentes. Temos comutação de circuitos, de pacotes e de mensagens. A comutação de circuitos é um método que permite que dois nós sempre estejam conectados através de um *link* dedicado. Na maioria dos casos, a comutação de circuitos é uma função da camada física. A comutação de pacotes é discutida no Capítulo 18 como uma particularidade da camada de enlace e no Capítulo 19 como uma especialidade da camada de rede.

Meios de Transmissão

A camada física é altamente dependente dos meios de transmissão para transportar *bits* na forma de sinais eletromagnéticos. A camada física controla a utilização do meio de transmissão,

embora os meios de transmissão sejam independentes da camada física. Ondas eletromagnéticas podem ser guiadas ou não-guiadas. Alguns exemplos de meios que servem de guias de onda são o par trançado, o cabo coaxial e o cabo de fibra óptica. Eles serão discutidos na seção que trata dos meios de transmissão. Os melhores exemplos de ondas não-guiadas incluem a comunicação via rádio e microondas.

Rede Telefônica

A maioria das redes nos dias de hoje inicia-se na rede telefônica. As redes telefônicas foram utilizadas durante muito tempo com o intuito único de prover a comunicação de voz ao redor do mundo. Quando há necessidade de comunicação de dados, muitas vezes a rede telefônica serve de fundamento. Os dados são transformados em sinais analógicos e são enviados através do canal de voz. Discutimos a rede telefônica como prelúdio para outros tipos específicos de redes de dados e também como um bom exemplo de uma rede cuja funcionalidade da camada física será descrita nesta parte do livro. Daremos também um breve histórico da rede telefônica para explicar as razões de desenvolvimentos de redes recentes tais como as LATAs.

Acesso em Alta Velocidade

Acessar a Internet requer uma conexão física entre o usuário final e um servidor de acesso (ISP). Introduziremos os modems como dispositivos de acesso à Internet. Em seguida, apresentaremos duas soluções tecnológicas alternativas: DSL e a TV a cabo. A tecnologia DSL proporciona uma conexão física de alta velocidade, outra vez utilizando as linhas telefônicas já existentes. A conexão via TV a cabo permite que o usuário utilize alguns canais anteriormente atribuídos à transmissão de *broadcasting* de vídeo para transferir dados para e da Internet.

Capítulos

A segunda parte do livro cobre sete capítulos. No Capítulo 3 são introduzidos os conceitos e características dos sinais como o veículo padrão de transporte de dados. Os Capítulos 4 e 5 apresentam as formas de conversão de sinais analógicos em digitais e vice-versa. O Capítulo 6 discute o conceito de multiplexação, um mecanismo importante na otimização da largura de banda de um meio de transmissão. Embora os meios de transmissão sejam a conexão permanente com a camada física dos dispositivos de rede, é a camada física que os controla. Deixamos para o Capítulo 7 o estudo dos meios de transmissão. O Capítulo 8 discute os processos de comutação, um tópico que pode envolver muitas camadas do modelo. Entretanto, iremos focar a técnica de comutação de circuitos, visto que é aquela que, a maioria das vezes, faz uso da camada física. Para mostrar uma aplicação da comutação de circuitos, introduziremos a rede telefônica e muitos outros tópicos relacionados a ela. Por fim, como o propósito principal da maioria das redes atuais é acessar a Internet, o Capítulo 9 apresenta muitas tecnologias de acesso à Internet.

Capítulo 3

Sinais

Uma das funções mais importantes da camada física é converter dados em sinais eletromagnéticos e transmiti-los através de um meio de transmissão. Não importa se você está fazendo aquisição de dados estatísticos de um outro computador, enviando figuras animadas de uma estação de trabalho de *design* ou se você está manipulando um banco de dados localizados num centro de controle distante, o processo de transmissão de dados está acontecendo através das conexões de rede.

Geralmente, os dados manipulados por um usuário ou *software* não estão numa forma adequada para serem transmitidos na rede. Por exemplo, você não consegue enrolar um fotografia, inseri-la num fio e transmiti-la através da rede. Entretanto, você pode transmitir uma descrição codificada da fotografia. Em vez de enviar a fotografia real, você usa algum mecanismo de codificação para gerar uma cadeia de *bits* 1s e 0s capaz de informar ao dispositivo receptor como reconstruir a imagem da fotografia.

Contudo, mesmo 1s e 0s não podem ser enviados através de *links* da rede. Antes, eles devem ser convertidos para uma forma que o meio de transmissão possa aceitá-los. Os meios de transporte trabalham conduzindo energia ao longo de um caminho físico. Assim, a primeira funcionalidade da camada física é converter as cadeias de 1s e 0s em sinais eletromagnéticos para o transporte de energia.

Para serem transmitidos, os dados devem ser convertidos em sinais eletromagnéticos.

3.1 ANALÓGICO E DIGITAL

Tanto dados como sinais que os representam podem existir na forma *análogica* ou *digital*.

Informação Analógica e Informação Digital

Talvez o melhor exemplo de representação de **informação analógica** seja a voz humana. Quando uma pessoa fala é gerada uma onda analógica no ar. Essa onda pode ser capturada por um microfone e convertida em um sinal analógico ou amostrada e convertida em um sinal digital.

Um bom exemplo de representação de **informação digital** são os dados armazenados na memória de um computador na forma de 0s e 1s. Quando provocamos uma transferência de dados de uma posição de memória para outro local de armazenamento, esses dados são convertidos em

um sinal digital, para dentro ou para fora do computador, ou são modulados a um sinal analógico e, então, são enviados através de um meio de transmissão para outro computador.

Sinais Analógicos e Sinais Digitais

Assim como a informação, os **sinais** podem ser representados na forma analógica ou digital. Um **sinal analógico** possui infinitos níveis de tensão num certo período de tempo. Quando uma onda evolui do valor A para o valor B, ela passa através de um número infinito de valores ao longo do caminho. Contrariamente, um **sinal digital** possui apenas um número limitado e definido de valores, simplificados freqüentemente como 1 e 0.

O modo mais simples de representar sinais é utilizando um plano cartesiano constituído de um par de eixos perpendiculares, onde o eixo vertical representa sempre o valor ou a intensidade do sinal e o eixo horizontal representa o fluxo do tempo. A Figura 3.1 ilustra dois sinais, um analógico e outro digital. A curva que representa o sinal analógico passa sempre através de um número infinito de pontos (valores). Entretanto, as linhas verticais do sinal digital demonstram apenas uma transição repentina (um degrau) que o sinal realiza entre dois valores.

Sinais podem ser analógicos ou digitais. Sinais analógicos possuem um número infinito de valores distribuídos numa faixa. Ao passo que os sinais digitais possuem apenas um número limitado de valores.

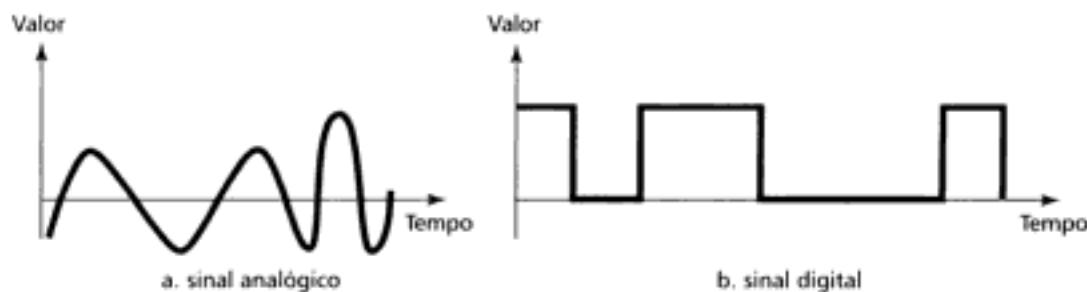


Figura 3.1 Comparação entre sinal analógico e sinal digital.

Sinais Periódicos e Sinais Não Periódicos

Tanto um sinal analógico quanto um sinal digital pode se apresentar na forma *periódica* ou *não periódica*. Um **sinal periódico** completa um padrão dentro de um intervalo de tempo mensurável, denominado **período**, e repete este padrão nos períodos de tempo subseqüentes. A um padrão completo é dado o nome de **ciclo**. Um sinal **não periódico** evolui no tempo sem exibir um padrão ou completar um ciclo.

Na comunicação de dados é freqüente o uso de sinais analógicos periódicos e sinais digitais não periódicos para enviar dados de um ponto a outro, apesar de ambos poderem assumir tanto a forma periódica quanto a não periódica.

Na comunicação de dados, utilizamos freqüentemente sinais analógicos periódicos e sinais digitais não periódicos.

3.2 SINAIS ANALÓGICOS

Sinais analógicos são classificados como sinais simples e sinais compostos. Um sinal analógico simples, uma **onda senoidal**, não pode ser decomposto numa soma simplificada de sinais. Um sinal analógico composto é constituído de uma soma discreta, possivelmente infinita, de múltiplas ondas senoidais.

A Onda Senoidal

A onda senoidal é a forma fundamental de um sinal analógico periódico de maior importância na comunicação de dados. Visualizado como uma curva variável no tempo, varia ao longo de um ciclo de forma contínua. A Figura 3.2 ilustra uma onda senoidal. Cada ciclo da senóide consiste de dois arcos da função seno, um acima e outro abaixo do eixo dos tempos.

Podemos descrever matematicamente uma onda senoidal como segue:

$$s(t) = A \operatorname{sen}(2\pi ft + \phi)$$

sendo s o valor instantâneo do sinal, A a amplitude de pico, f a freqüência e ϕ a fase da onda. Essas características descrevem completamente uma onda senoidal.



Figura 3.2 Uma onda senoidal.

Amplitude de pico

A **amplitude de pico** de um sinal representa o valor de intensidade mais alta, proporcionalmente à energia transportada pelo sinal. Para sinais elétricos, a amplitude de pico geralmente é medida em *volts* (veja a Fig. 3.3).

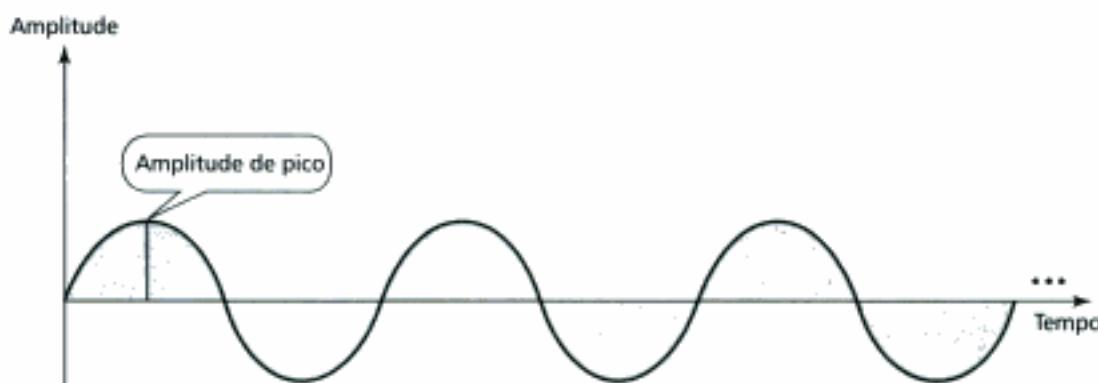


Figura 3.3 Amplitude.

Período e Freqüência

Período é o intervalo de tempo que uma onda leva para completar um ciclo. Assim, período é uma grandeza temporal, logo medido em segundos. A **freqüência** é o número de períodos ou ciclos num intervalo de tempo igual a 1 segundo. Perceba que período e freqüência representam a mesma característica de uma onda definida de duas formas diferentes. O período é o inverso da freqüência, como mostram as fórmulas abaixo.

$$f = \frac{1}{T} \quad \text{e} \quad T = \frac{1}{f}$$

Período e freqüência são grandezas inversamente proporcionais.

A Figura 3.4 ilustra os conceitos de período e freqüência.

O período de uma onda é expresso formalmente em segundos, enquanto que a freqüência é expressa em **hertz (Hz)**, como mostra a Tabela 3.1.

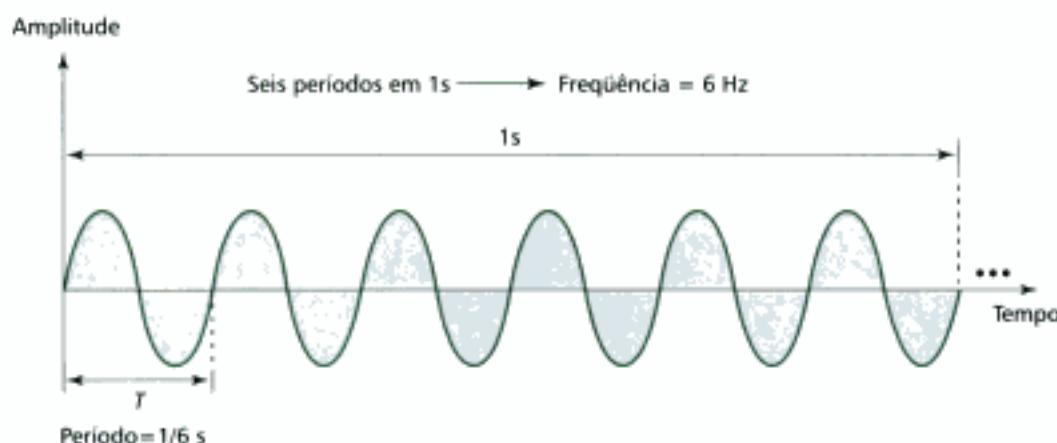


Figura 3.4 Período e freqüência.

TABELA 3.1 Unidades de período e freqüência

Unidade	Equivalência	Unidade	Equivalência
Segundos (s)	1 s	hertz (Hz)	1 Hz
Milissegundos (ms)	10^{-3} s	kilohertz (kHz)	10^3 Hz
Microssegundos (μs)	10^{-6} s	megahertz (MHz)	10^6 Hz
Nanosegundos (ns)	10^{-9} s	gigahertz (GHz)	10^9 Hz
Picossegundos (ps)	10^{-12} s	terahertz (THz)	10^{12} Hz

Exemplo 1

Expressar um período de 100ms em microssegundos e determinar a freqüência correspondente em kilohertz.

Solução

Primeiramente, vamos expressar 100ms em microssegundos. Da Tabela 3.1, encontramos o equivalente de 1ms ($1\text{ms} = 10^{-3}\text{s}$) e 1s ($1\text{s} = 10^6\text{μs}$). Encontramos o resultado desejado tomando as seguintes substituições:

$$100\text{ms} = 100 \times 10^{-3}\text{s} = 100 \times 10^{-3} \times 10^6\text{μs} = 10^5\text{μs}$$

Agora, usamos a relação de reciprocidade entre freqüência e período, e convertemos o resultado para kilohertz ($1\text{Hz} = 10^3\text{kHz}$).

$$100\text{ms} = 100 \times 10^{-3}\text{s} = 10^{-1}\text{s} \rightarrow f = \frac{1}{10^{-1}}\text{Hz} = 10 \times 10^3\text{kHz} = 10^4\text{kHz}$$

Um Pouco Mais Sobre Freqüência

Sabemos que período relaciona um sinal com o tempo e que a freqüência de uma onda é o número de ciclos completos por segundo. Podemos olhar de uma forma diferente e encará-la como sendo uma medida de taxa de variação. Sinais eletromagnéticos são formas de onda variáveis no tempo, isto é, são flutuações contínuas preditas, acima e abaixo de um nível médio de energia. Basta

ver que um sinal de 40Hz tem a metade da freqüência de um sinal de 80Hz. Assim, no tempo de um único ciclo do sinal de 40Hz são realizados dois ciclos do sinal de 80Hz. Então, cada ciclo do sinal de 40Hz também leva um tempo duas vezes maior para mudar da maior amplitude positiva para a amplitude mais negativa. A conclusão que tiramos disso é que, embora a freqüência descreva o número de ciclos por segundo (hertz), ela pode ser vista como uma medida genérica da taxa de variação de um sinal com relação ao tempo.

Além disso, se o valor de um sinal muda num curto espaço de tempo, a freqüência desse sinal é elevada. Se as mudanças nos valores ocorrem apenas em grandes intervalos de tempo, a freqüência desse sinal é baixa.

Freqüência é a taxa de variação com relação ao tempo. Variações curtas no tempo indicam que o sinal possui freqüência alta. Variações longas no tempo indicam que o sinal possui freqüência baixa.

Dois Extremos

E se o sinal não variar no tempo? Ou, se o nível de tensão se mantém constante por um longo intervalo de tempo? Em tais casos, o sinal tem freqüência zero. Conceitualmente, a idéia é simples. Se um sinal não varia de forma alguma, ele nunca completa um ciclo. Logo deve possuir freqüência 0Hz.

E se o sinal variar instantaneamente? Ou, e se o sinal mudar de um nível de tensão a outro em um tempo zero? Então, a melhor resposta seria que esse sinal possui freqüência infinita. Em outras palavras, quando um sinal varia instantaneamente, o período correspondente é zero. Como a freqüência é o recíproco do período, nesse caso, a freqüência seria 1/0. Essa divisão é algo inimaginável, que acreditamos ser ilimitada, e que pressupomos representar uma freqüência infinita.

Se um sinal é constante no tempo, a freqüência correspondente é zero. Se um sinal variar instantaneamente no tempo, a freqüência correspondente tende a infinita.

Fase

O termo **fase** descreve a posição da forma de onda com relação ao marco zero do tempo. Se imaginarmos uma onda como algo que pode ser deslocado para frente e para trás no eixo dos tempos, a fase descreve o quanto um sinal está deslocado em relação ao tempo zero. Podemos dizer que a fase indica o *status* do primeiro ciclo.

A fase descreve a posição de uma forma de onda relativa ao tempo zero.

A fase é medida em graus ou radianos [$360^\circ = 2\pi$ rad; $1^\circ = (2\pi/360)$ rad e 1 rad = $360/(2\pi)$]. Um deslocamento de fase de 360° corresponde a deslocar um período completo da onda. Já um deslocamento de 180° corresponde a um deslocamento de meio ciclo (semi-ciclo) de período. Por fim, um deslocamento de 90° corresponde a um deslocamento de 1/4 de período (veja Fig. 3.5).

Exemplo 2

Uma senóide está posicionada a 1/6 de um ciclo com relação ao tempo zero. Qual é o deslocamento de fase em graus e em radianos?

Solução

Sabemos que um ciclo completo representa 360° . Sendo assim, 1/6 de ciclo é

$$\frac{1}{6} \times 360^\circ = 60^\circ = 60 \times \frac{2\pi}{360} \text{ rad} = 1,046 \text{ rad}$$

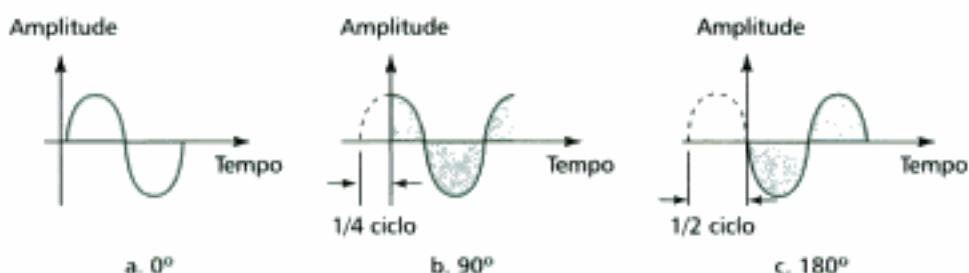


Figura 3.5 Relação entre diferentes fases.

Exemplos de Ondas Senoidais

Uma comparação visual entre sinais de diferentes características pode dar uma melhor compreensão destas características. A Figura 3.6 mostra três ondas senoidais com diferentes amplitudes de pico, freqüências e fases.

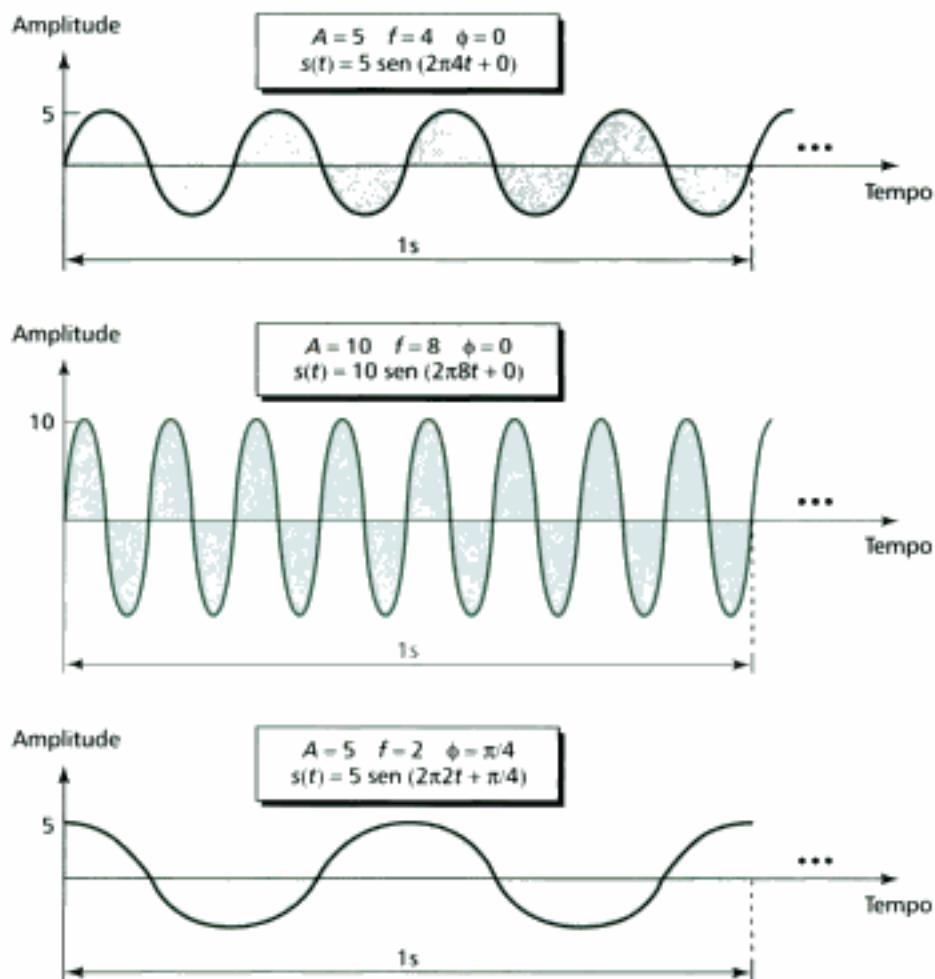


Figura 3.6 Exemplos de ondas senoidais.

Domínio do Tempo versus Domínio da Freqüência

Uma onda senoidal fica bem definida através da amplitude, freqüência e fase. Temos mostrado graficamente ondas senoidais esboçadas usando o que é chamado **domínio do tempo**. Um gráfico no domínio do tempo mostra as variações instantâneas de um sinal. A fase e a freqüência não são medidas explícitas em um gráfico no domínio do tempo.

Para mostrar a relação entre amplitude e freqüência de um sinal podemos usar o que é denominado **domínio da freqüência**. A Figura 3.7 compara o domínio do tempo (valores instantâneos) e o domínio da freqüência (amplitude de pico com relação à freqüência).

A figura mostra três sinais de freqüências diferentes. Compare cada par de resultados colados lado a lado para ter uma idéia que tipo de dados cada uma das representações pode oferecer. Perceba que todos os três sinais têm amplitude de 5volts (5V). A freqüência do primeiro sinal é 0. No domínio da freqüência mostramos uma resposta (um *spike*) no ponto de freqüência 0 e amplitude 5. O segundo sinal tem freqüência 8. Assim, exibimos no domínio da freqüência uma resposta (*spike*) de amplitude 5 no ponto de freqüência 8. Finalmente, o terceiro sinal é mostrado com uma freqüência 16 e com a mesma amplitude dos demais. Perceba que no domínio da freqüência podemos mostrar duas características de um sinal através de um único *spike*; a posição no eixo horizontal representando a freqüência e a altura o valor de amplitude de pico. A fase de um sinal não pode ser exibida no domínio da freqüência. Para fazê-lo, necessitamos de um outro domínio que não é discutido neste livro.

A melhor forma de representar um sinal analógico é usando o domínio da freqüência.

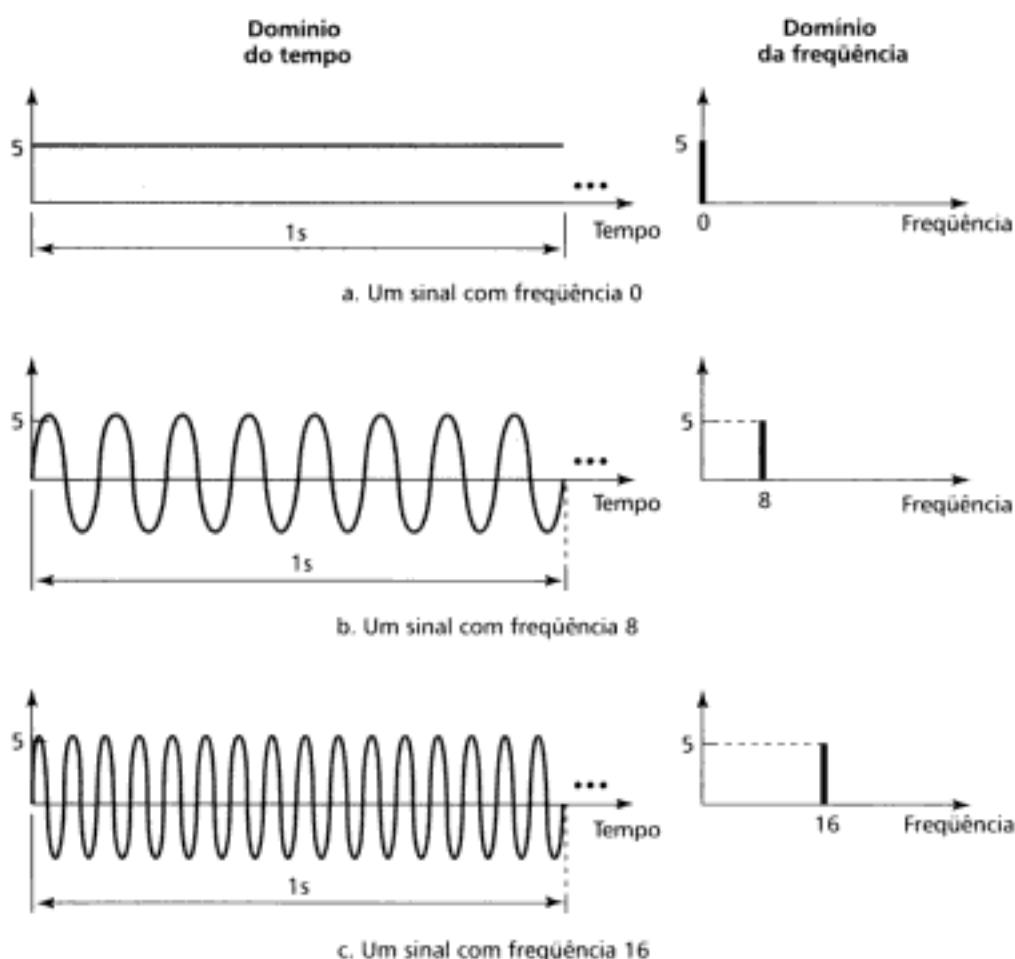


Figura 3.7 Domínio do tempo versus da freqüência.

Sinais Compostos

Até aqui, focamos nossa atenção na discussão de sinais simples (ondas senoidais). Embora uma única onda senoidal seja útil para alguns propósitos, ela encontra pouca utilidade na comunicação de dados. Podemos enviar uma onda senoidal de freqüência 60Hz através de uma linha de dis-

tribuição de energia elétrica para alimentar nossas casas, empresas, etc. Podemos ainda, usar uma única senóide para enviar um alarme a um centro de segurança quando um ladrão arromba uma porta ou uma janela da nossa casa. No primeiro caso, a senóide transporta energia. No segundo, a presença do sinal leva o centro de segurança a inferir o perigo.

Se utilizássemos uma única senóide para transportar a conversação numa linha telefônica, a informação transmitida e recebida seria semelhante ao som de uma cigarra ou campainha elétrica. Não haveria sentido e nem muito menos transporte de informação. Se enviássemos uma única onda senoidal para transmitir dados, estaríamos sempre trabalhando no regime do tudo ou nada, como se estivéssemos enviando 1s (tudo) ou 0s (nada)*, a qual não estabelece comunicação inteligível.

Um sinal de uma única freqüência não é útil aos propósitos da comunicação de dados. Necessitamos mudar uma ou mais características do sinal para torná-lo útil.

Se quisermos utilizar uma onda senoidal para comunicação, necessitamos modificar pelo menos uma das características dela. Por exemplo, quando os dados a serem enviados são o *bit* 1, podemos enviar o valor máximo de tensão (amplitude de pico). Quando o *bit* a ser enviado é o 0, podemos enviar o valor mais negativo (amplitude mínima). Contudo, devemos sempre ter em mente que quando mudamos alguma característica de uma onda senoidal, não estamos mais lidando com uma onda senoidal simples. Em vez disso, passamos a lidar com um **sinal composto** construído a partir de inúmeras ondas senoidais. A mais sutil mudança na amplitude, freqüência ou fase da senóide gera um novo conjunto ou espectro de freqüências. Intuitivamente, mudança (variação) está relacionada à freqüência. Quanto mais mudanças forem criadas num sinal, maior a quantidade de novas freqüências criadas.

Quando modificamos qualquer uma das características de um sinal simples associamos ao novo sinal outras componentes de freqüência e, assim, o transformamos em um sinal composto.

Análise de Fourier

No início do século XIX, o matemático francês Jean-Baptiste Fourier mostrou que qualquer sinal composto é a soma de um conjunto de senóides de diferentes freqüências, fases e amplitudes. Em outras palavras, podemos escrever matematicamente um sinal composto como:

$$s(t) = A_1 \operatorname{sen}(2\pi f_1 t + \phi_1) + A_2 \operatorname{sen}(2\pi f_2 t + \phi_2) + A_3 \operatorname{sen}(2\pi f_3 t + \phi_3) + \dots$$

De acordo com a análise de Fourier, qualquer sinal composto pode ser representado por uma combinação de senóides simples de diferentes freqüências, amplitudes e fases.

Por exemplo, vamos considerar o caso clássico de uma onda quadrada de amplitude A e freqüência f (período T) da Figura 3.8. Com base na **análise de Fourier**, é possível provar que esse sinal pode ser decomposto na série de ondas senoidais mostrada a seguir.

$$s(t) = \operatorname{sen}[2\pi f t] + \operatorname{sen}[2\pi(3f)t] + \operatorname{sen}[2\pi(5f)t] + \dots$$

Em outras palavras, essa é a série de senos cujas freqüências são $f, 3f, 5f, 7f, \dots$ e amplitudes são $4A/\pi, 4A/3\pi, 4A/5\pi, 4A/7\pi$ e assim por diante. O termo dominante, ou seja, de maior amplitude na série, é aquele de freqüência f , denominado **freqüência fundamental**. Os termos de freqüência $3f, 5f, 7f, \dots$ são denominados, respectivamente, de terceiro, quinto e sétimo harmônicos e

* N. de R. T.: Vale lembrar que o código Morse funciona exatamente desta forma: tudo ou nada em intervalos periódicos. Entretanto, este tipo de transmissão é muito rudimentar e encontra pouco uso prático.

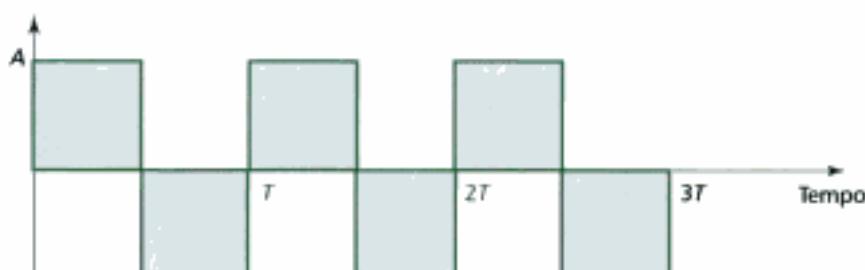


Figura 3.8 Onda quadrada.

assim por diante. Desse modo, para recriar um sinal quadrado completo é necessário somar todos os harmônicos de freqüência ímpares até infinito*. Por exemplo, se uma onda quadrada possui freqüência 5kHz, os componentes da série tem freqüências 5kHz, 15kHz, 25kHz,... A Figura 3.9 mostra a fundamental e dois outros harmônicos.

É claro que, se somarmos a fundamental e os dois harmônicos, não reconstruímos a onda quadrada original, mas o padrão resultante leva-nos a intuir que mais alguns harmônicos são suficientes para conseguir uma boa aproximação (ver Figura 3.10).

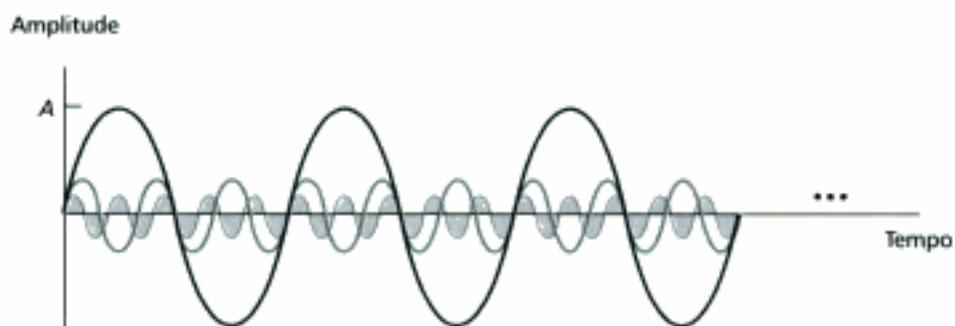


Figura 3.9 Três harmônicos.

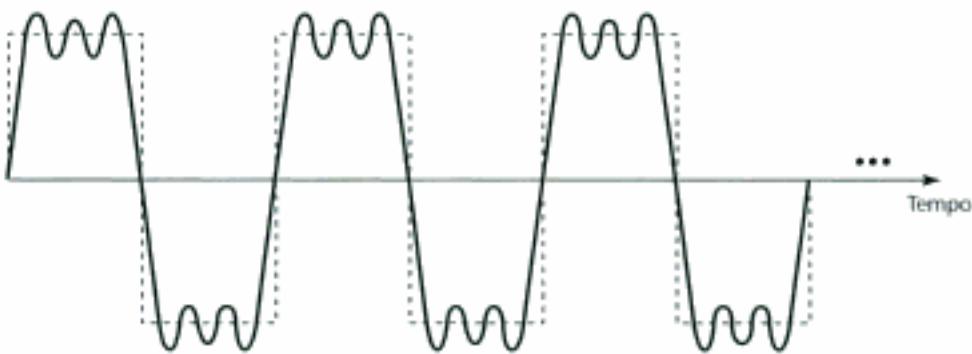


Figura 3.10 Adicionando primeiro três harmônicos.

* N. de R. T.: Tecnicamente, não temos condições de lidar com os infinitos termos da série. Por isso, utilizamos a idéia de convergência da série para aproximar um sinal composto a partir de uma soma finita de termos. Por exemplo, uma boa aproximação da onda quadrada pode ser obtida com 10 termos (a fundamental + 9 harmônicos de freqüência ímpares).

Espectro de Freqüência

A descrição completa de um sinal composto no domínio da freqüência é denominado **espectro de freqüência** desse sinal. Por exemplo, a Figura 3.11 apresenta os espectros de freqüência de uma onda quadrada e de um sinal que se aproxima da onda quadrada (fundamental + dois harmônicos), respectivamente.

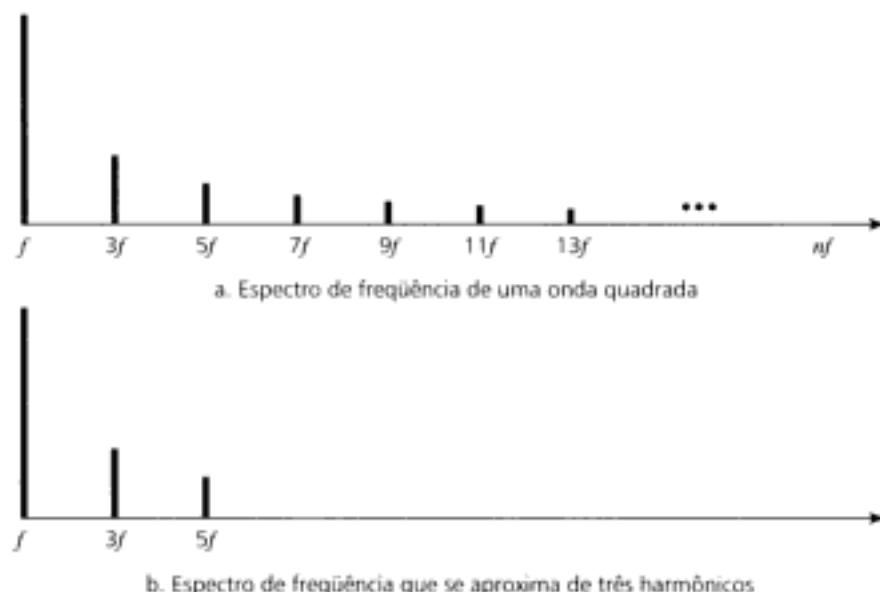


Figura 3.11 Comparação do espectro de freqüência.

Sinal Composto e o Meio de Transmissão

Fisicamente, um sinal viaja através de um meio de transmissão de suporte (cabo ou ar). Contudo, cada meio possui características próprias e que não dependem da existência de um sinal. Uma das características de um meio está relacionada às freqüências que ele pode transmitir. Desse modo, um meio pode facilmente transmitir algumas freqüências ou bloquear (filtrar) outras. Isto significa que, quando um sinal composto de muitas freqüências é enviado através de uma extremidade de um meio de transmissão, pode ser que o sinal recebido na outra extremidade não seja o mesmo sinal original. Para assegurar a integridade do sinal composto, o meio deve permitir a passagem de quaisquer freqüências, além disso, deve preservar as amplitudes e fases de cada um dos harmônicos.

Estamos querendo dizer que nenhum meio de transmissão é perfeito. Cada tipo de meio permite a passagem de algumas freqüências, atenua outras e, até mesmo, bloqueia completamente certos harmônicos em muitos casos. Em particular, significa que ao enviarmos uma onda quadrada através de um meio é comum obtermos algo na extremidade de saída do receptor diferente da onda quadrada original. A Figura 3.12 ilustra esse conceito.

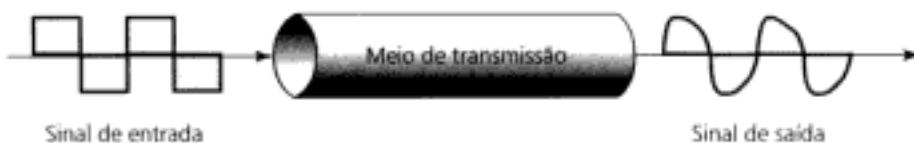


Figura 3.12 Sinal corrompido.

Largura de Banda

A faixa de freqüências passantes de um meio é denominada **largura de banda (bandwidth)**. Porque nenhum meio é capaz de passar ou bloquear todas as freqüências, a largura de banda normalmente refere-se à faixa de freqüências que o meio pode transmitir com perdas abaixo da metade da

potência inicial do sinal. Assim, a largura de banda é uma quantidade que normalmente se refere à diferença entre duas freqüências. Por exemplo, se um meio permite a passagem de sinais de freqüência entre 1kHz e 5kHz, com perdas abaixo da metade da potência inicial do sinal, a largura de banda vale $5\text{kHz} - 1\text{kHz} = 4\text{kHz}$.

A largura de banda é uma propriedade de um meio. Ela é a diferença entre a maior e a menor freqüências que um meio pode transmitir satisfatoriamente.

Se a largura de banda de um meio não casa com o espectro de freqüências de sinal, fatalmente algumas freqüências do sinal serão perdidas durante a transmissão. Por exemplo, a onda quadrada da Figura 3.8 tem freqüências que se expandem até o infinito. Nenhum meio de transmissão tem uma largura de banda capaz de transmiti-la integralmente. Isto significa que uma onda quadrada passando através de um meio sofrerá sempre alguma deformação no sinal. Um exemplo mais, a voz humana normalmente tem um espectro variando entre 300 e 3300Hz (largura de banda de 3000Hz). Se usarmos uma linha de transmissão com largura de banda de 1000Hz, entre 1500 e 2500Hz, é inevitável que alguns componentes de freqüência do sinal de voz sejam perdidos ao ponto de torná-la irreconhecível.

Algumas pessoas usam o termo *largura de banda* para classificar um sinal. Por exemplo, é comum ouvir "Este sinal possui uma largura de banda de 1kHz". Neste caso, eles querem dizer que o sinal possui um espectro de freqüência cujas freqüências predominantes são perto de 1kHz. Dito de outra maneira, seria "Precisamos de um meio com largura de banda de 1kHz se quisermos enviar este sinal sem perdas significativas dentro dele". Hoje em dia, as pessoas se referem indistintamente à largura de banda do meio e dos sinais, mas nem sempre foi assim.

Neste livro, usaremos o termo *largura de banda* referindo-nos à propriedade de um meio ou à largura do espectro de um sinal.

A Figura 3.13 apresenta visualmente o conceito de largura de banda. A figura ilustra a faixa de freqüências que um meio permite a passagem e as amplitudes relativas das freqüências passantes. Perceba que o meio pode passar freqüências acima de 5kHz ou abaixo de 1kHz, mas de acordo com o critério de meia potência anterior, as amplitudes de potência dos sinais com essas freqüências são menores que a metade da potência original do sinal, logo são consideradas bloqueadas.

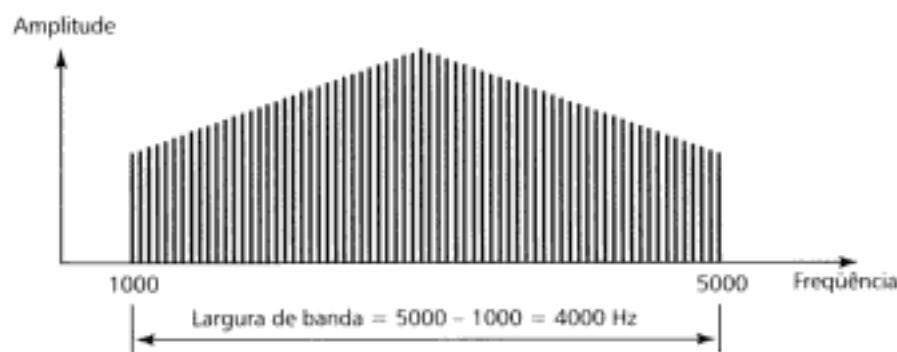


Figura 3.13 Largura de banda.

Exemplo 3

Qual é a largura de banda de um sinal periódico decomposto em cinco componentes senoidais de freqüências 100, 300, 500, 700 e 900Hz? Desenhe o espectro de freqüências levando em conta que todas as componentes têm a mesma amplitude de pico em 10V.

Solução

Seja f_h a maior e f_l a menor freqüência. Considere ainda que a largura de banda seja representada por B . Então,

$$B = f_h - f_l = 900 - 100 = 800 \text{ Hz}$$

O espectro de freqüências possui somente cinco *spikes* localizados em 100, 300, 500, 700 e 900Hz (veja Fig. 3.14).

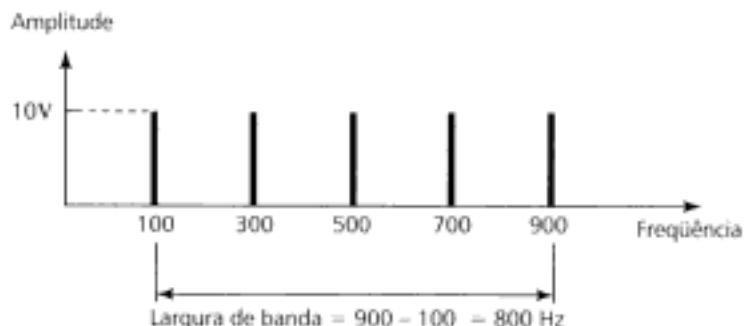


Figura 3.14 Exemplo 3.

Exemplo 4

Um sinal composto possui uma largura de banda de 20Hz. Sabendo que a maior freqüência vale 60Hz, qual é a menor freqüência que constitui esse sinal? Desenhe o espectro de freqüência considerando que o sinal contém todas as freqüências inteiros e de mesma amplitude entre a menor e a maior freqüências.

Solução

Seja f_h a maior freqüência e f_l a menor freqüência. Considere ainda que a largura de banda seja representada por B . Então,

$$\begin{aligned} B &= f_h - f_l \\ 20 &= 60 - f_l \\ f_l &= 60 - 20 = 40 \text{ Hz} \end{aligned}$$

O espectro de freqüências inicia em 40Hz e estende-se até 60Hz, exibindo todas as freqüências inteiros nessa faixa. Na Figura 3.15, mostramos isso através de uma série de *spikes*.

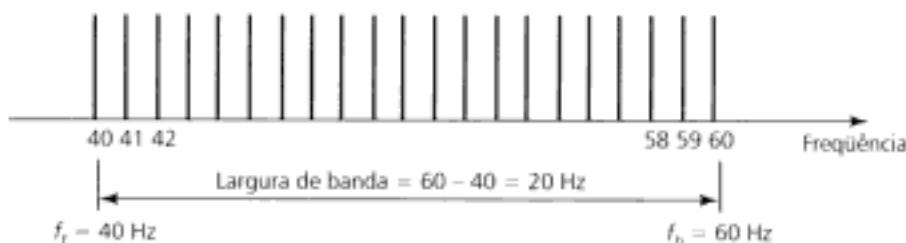


Figura 3.15 Exemplo 4.

Exemplo 5

Um sinal possui um espectro de freqüência que vai de 1 a 2kHz (largura de banda = 1kHz). Um meio pode transmitir freqüências compreendidas na faixa que vai de 3 a 4kHz (largura de banda = 1kHz). Este sinal consegue viajar através desse meio?

Solução

A resposta é definitivamente não. Embora o sinal tenha a mesma largura de banda do meio (1kHz), as faixas de freqüência não se sobreponham. O meio só pode transmitir freqüências entre 3 e 4kHz. Nessa faixa, o sinal é totalmente perdido.

3.3 SINAIS DIGITAIS

Além da representação analógica, um sinal também pode possuir uma forma de representação digital. Nessa representação, em geral, o nível 1 equivale a uma tensão positiva e o nível 0 equivale ao referencial de zero volt (veja Fig. 3.16).

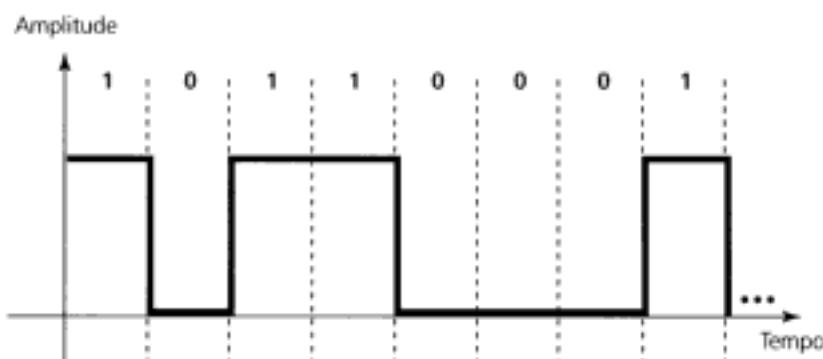


Figura 3.16 Um sinal digital.

Intervalo de Sinalização e Número de Bits por Segundo

A maioria dos sinais digitais não são periódicos. Sendo assim, os termos período e freqüência não são apropriados. Dois novos termos, *intervalo de sinalização* (no lugar de período) e *número de bits por segundo* (no lugar de freqüência) são utilizados para descrever sinais digitais. O **intervalo de sinalização** é o tempo necessário para enviar um único bit. O **número de bits por segundo** é a quantidade de intervalos de sinalização por segundo. Isto significa que o número de bits por segundo é a quantidade de bits enviados num tempo igual a 1s, usualmente expresso por **bits por segundo (bps)** (veja Figura 3.17).

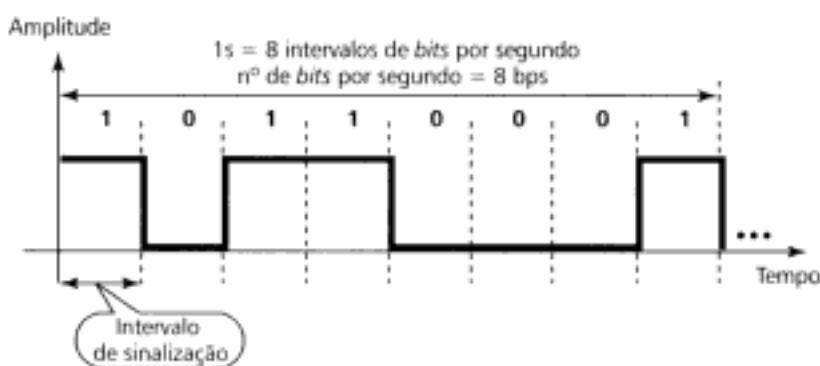


Figura 3.17 Intervalo de sinalização e número de bits por segundo.

Exemplo 6

Um sinal digital possui um número de bits por segundo de 2000bps. Qual é a duração de cada bit, ou seja, o intervalo de sinalização?

Solução

O intervalo de sinalização é o recíproco do número de bits por segundo. Logo,

$$\text{Intervalo de sinalização} = \frac{1}{\text{Número de bits por segundo}} = \frac{1}{2000} = 0,0005\text{s} = 0,0005 \times 10^6 \mu\text{s} = 500\mu\text{s}$$

Sinal Digital como um Sinal Analógico Composto

Deve ficar claro por enquanto que um sinal digital, com tantas mudanças bruscas, é de fato um sinal composto de um número infinito de freqüências. Em outras palavras, a largura de banda de um sinal digital é infinita.

Um sinal digital é um sinal composto de largura de banda infinita.

Sinal Digital em um Meio Banda Larga

Se um meio possui uma largura de banda larga, mas finita, podemos enviar um sinal digital através dele. Claro que muitas freqüências serão bloqueadas pelo meio de transmissão, mas uma faixa de freqüências passantes existentes no sinal ainda deve ser suficiente para preservar decentemente a forma do sinal digital. Veremos adiante que é possível utilizar um meio dedicado, tal como um cabo coaxial, para enviar um sinal digital através de uma rede local até algumas centenas de metros sem repetidor.

Sinal Digital em um Meio de Largura de Banda Limitada

Podemos enviar um sinal digital através de um meio de transmissão de largura de banda limitada? A resposta para esta questão é definitivamente sim. Todos os dias, enviamos dados através do canal de voz (as linhas telefônicas) para a Internet. Mas, qual é a largura de banda (B) mínima, em hertz, necessária para enviar n bps? Dito de outra forma, qual é a relação entre o número de *bits* por segundo e a largura de banda de um meio? Estas questões serão respondidas formalmente quando discutirmos o teorema de Nyquist e a lei de Shannon para a capacidade de transmissão máxima de um meio. Nesta seção, faremos uma aproximação com o intuito de preparamos o cenário para que os fundamentos da transmissão de dados sejam compreendidos.

Usando Apenas um Harmônico

Para simplificar a discussão, imagine que dispomos de um computador capaz de transmitir a apenas 6bps. Esta situação é hipotética de maneira a possibilitar a visualização dos resultados graficamente. Em cada segundo, o computador produz 6 *bits*. Num segundo podemos ter 111111, noutro 001010, depois 101010 e assim por diante. Representaremos o nível 1 com um valor de amplitude positiva e um valor de amplitude negativa representará o 0. A Figura 3.18 mostra dois sinais.

Vejamos se conseguimos simular qualquer um desses padrões utilizando um sinal com uma única freqüência. Os melhores casos são o 111111 ou 000000. Nesses casos, podemos enviar um sinal de freqüência zero. Os piores casos são definitivamente 101010 ou 010101. Eles são os piores casos porque, dentre todas as possíveis combinações de 1s e 0s em 6 *bits*, são os casos onde ocorrem o maior número de variações de 0 para 1 e vice-versa. Quanto maior o número de variações abruptas no sinal, maiores são as freqüências que o compõem. Entretanto, podemos simular este sinal digital usando um sinal analógico de freqüência 3Hz, ou seja, com a metade do número de *bits* por segundo. Assim, temos:

Melhor caso:	Número de <i>bits</i> por segundo = 6	freqüência = 0Hz
Pior caso:	Número de <i>bits</i> por segundo = 6	freqüência = 3Hz

Podemos verificar que todas as outras situações estão entre o melhor e pior caso. Desse modo, podemos simular os demais casos usando uma freqüência apenas, 1 ou 2 Hz, e utilizando a relação de fase apropriada.

Noutras palavras, se for necessário simular a transmissão deste sinal digital nessa taxa de 6bps, será necessário, às vezes, enviar um sinal de freqüência 0Hz, outras vezes de freqüência 1Hz,

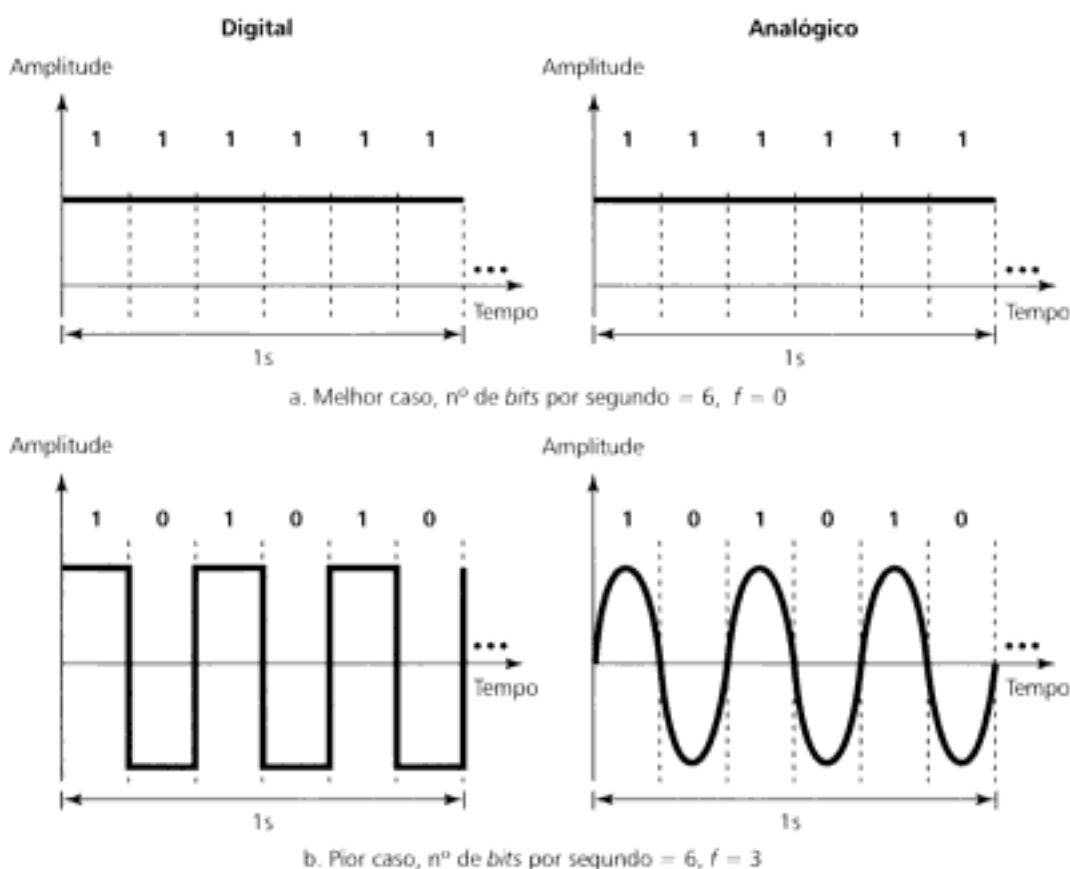


Figura 3.18 Digital versus analógico.

algumas vezes de 2Hz e outras de 3Hz. É claro que o meio utilizado para essa transmissão também deverá ser capaz de transmitir de 0 a 3Hz. Dessa forma, esse meio deverá ter uma largura de banda de 3Hz.

Se generalizarmos este exemplo simples, chegaremos a uma relação extremamente simples entre o número de *bits* por segundo e a largura de banda do meio. Para enviar n *bits* através de um canal analógico utilizando a aproximação anterior, necessitaremos de uma largura de banda tal que:

$$B = \frac{n}{2}$$

Usando Mais Harmônicos

A discussão anterior foi baseada num único harmônico. Para cada padrão predeterminado enviamos um sinal de única freqüência entre 0 e 3Hz. Porém, em muitas situações, enviar um sinal de uma única freqüência não é muito apropriado; o sinal analógico enviado pode parecer muito diferente do sinal digital pretendido e o receptor pode não reconhecer o padrão corretamente.

Para ajustar a forma do sinal e melhorar o processo de comunicação, particularmente em altas velocidades, necessitamos adicionar mais harmônicos ao sinal. Parece claro da discussão anterior que há necessidade de adicionarmos mais harmônicos de ordem ímpar para a transmissão de um sinal digital. Se adicionarmos o terceiro harmônico em cada caso, a largura de banda necessária seria: $B = n/2 + 3n/2 = 4n/2$ Hz. Se adicionarmos o terceiro e quinto harmônicos, necessitaremos de uma largura $B = n/2 + 3n/2 + 5n/2 = 9n/2$ Hz e assim por diante. Noutras palavras, temos:

$$B \geq \frac{n}{2} \quad \text{ou} \quad n \leq 2B$$

A Tabela 3.2 mostra quais as larguras de banda necessárias para enviar 1Kbps utilizando este método.

Enfatizamos o seguinte: nesse método, assim como em outros, a largura de banda necessária para a transmissão é proporcional ao número de *bits* por segundo. Se dobrarmos o número de *bits*, precisaremos dobrar a largura de banda.

O número de *bits* e a largura de banda são proporcionais entre si.

TABELA 3.2 Requisitos de largura de banda

<i>Nº de bits por segundo</i>	<i>Harmônico 1</i>	<i>Harmônico 1, 3</i>	<i>Harmônico 1, 3, 5</i>	<i>Harmônico 1, 3, 5, 7</i>
1 Kbps	500 Hz	2 KHz	4,5 KHz	8 KHz
10 Kbps	5 KHz	20 KHz	45 KHz	80 KHz
100 Kbps	50 KHz	200 KHz	450 KHz	800 KHz

Largura de Banda Analógica versus Largura de Banda Digital

Toda a discussão anterior sobre a proporcionalidade entre a largura de banda e o número de *bits* conduz à idéia de largura de banda digital. Se estivermos enviando informação na forma analógica através de um meio, estaremos lidando com a largura de banda analógica desse meio (expressa em hertz). Agora, se estivermos enviando dados digitais através do meio, deveremos lidar com a largura de banda digital (expressa em *bits* por segundo). A largura de banda analógica é a faixa de freqüências que um meio permite a passagem. A largura de banda digital é o número máximo de *bits* por segundo que um meio pode transmitir. Em síntese, as duas representam a mesma propriedade do meio, mas diferem em escalas e unidades.

A largura de banda analógica de um meio é expressa em hertz e a largura de banda digital em *bits* por segundo.

Altas Taxas de Transmissão em Bits por Segundo

Na discussão anterior, alguns leitores podem ter ficado intrigados, especialmente se considerarmos a transmissão de dados cotidiana através do canal de voz. O canal de voz é utilizado pelos usuários comuns e possui uma largura de banda entre 3 e 4kHz. Entretanto, sabemos que às vezes enviamos (e recebemos) dados a velocidades bem maiores que 30kbps (utilizando um modem tradicional). De acordo com a nossa discussão, não deveríamos ser capazes de enviar mais de 8Kbps através do canal de voz! Então, o que há de errado na discussão anterior? Bem, não há nada errado na discussão anterior. O fato básico, não mencionado até aqui, é que os modems utilizam alguma técnica de modulação que permite a representação de múltiplos *bits* num único período de um sinal analógico. Discutiremos a fundo estas técnicas no Capítulo 5.

3.4 ANALÓGICO VERSUS DIGITAL

Chegamos finalmente à questão fundamental: devemos utilizar um sinal analógico ou um sinal digital? A melhor resposta a esta pergunta depende da situação e da largura de banda disponível no meio.

Passa-Baixas versus Passa-Banda (ou Passa-Faixa)

Um canal ou *link* de transmissão é um filtro passa-baixas ou passa-banda. Um **canal passa-baixas** permite a passagem de freqüências compreendidas entre 0 e f . Assim, nesse canal, o limite inferior de freqüência vale 0 e o limite superior pode ser qualquer freqüência (incluindo, possivelmente, o infinito). De outro modo, um **canal passa-banda** possui uma largura de banda compreendida entre as freqüências f_1 e f_2 . A Figura 3.19 mostra as larguras de banda dos dois tipos de canais.

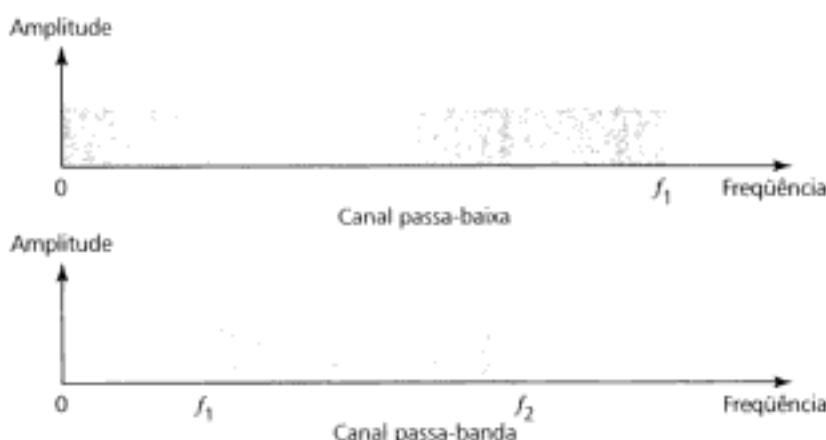


Figura 3.19 Passa-baixa versus passa-banda (ou passa-faixa).

Transmissão Digital

Um sinal digital precisa, teoricamente, de uma largura de banda infinita. O limite inferior (0Hz) é fixo e o limite superior (infinito) pode ir até onde nosso limite de aceitação permitir. É claro que, definindo o limite finito, o número de harmônicos fica limitado. No caso do canal passa-baixas isso significa uma largura de banda entre 0 e f .

Em geral, um canal passa-baixas é utilizado somente se o meio é dedicado entre dois dispositivos (ponto a ponto) ou compartilhado entre diversos dispositivos no tempo. Por exemplo, numa rede local coaxial, um cabo pode ser compartilhado entre as estações. Desse modo, nesse sistema, podemos transmitir dados digitalmente.

A transmissão digital necessita de um canal passa-baixas.

Transmissão Analógica

Um sinal analógico normalmente possui uma banda mais estreita que um sinal digital com freqüências entre f_1 e f_2 . Dito de outra forma, um sinal analógico requer um canal passa-banda. Além disso, a largura de banda de um sinal analógico pode ser deslocada a nosso bel-prazer. Por exemplo, podemos sempre deslocar um sinal com uma largura de banda entre f_1 e f_2 para um sinal com uma largura de banda entre f_3 e f_4 , mantendo a mesma largura de banda inicial.

Um canal passa-banda é mais comumente encontrado que um canal passa-baixas. A largura de banda de um meio pode ser dividida noutros canais passa-banda para transportar diversas transmissões analógicas. Por exemplo, numa analogia à telefonia celular, uma largura de banda limitada é dividida entre diversos usuários de telefone celular. Cada usuário possui uma largura entre 0 e 30kHz, com cada sinal deslocado apropriadamente.

A transmissão analógica pode usar um canal passa-banda.

Isto não significa que uma transmissão analógica não possa utilizar um canal passa-baixas. Significa apenas que, em geral, ela usa os canais mais disponíveis para transmissão: os canais passa-banda. Por fim, um canal passa-baixas é um caso especial de um canal passa-banda com $f_l = 0$.

3.5 LIMITES PARA A TAXA DE TRANSMISSÃO DE DADOS

Uma questão suprajacente na transmissão de dados é: o quanto rápido podemos enviar dados, em *bits* por segundo, através de um canal? A taxa de transmissão de dados depende de três fatores:

1. A largura de banda disponível
2. Os níveis de sinais que podemos utilizar
3. A qualidade do canal (o nível de ruído inerente ao canal)

Dois resultados teóricos foram desenvolvidos para determinar a taxa de transmissão de dados: um por Nyquist para um canal livre de ruído, outro por Shannon para um canal na presença de ruído.

Canal Livre de Ruídos: Fórmula para o Número de Bits por Segundo de Nyquist

Para um canal livre de ruídos, a **fórmula de Nyquist** determina o valor teórico máximo para a capacidade de transmissão de um meio, em *bits* por segundo:

$$C_N = 2 \times B \times \log_2 L$$

Nesta fórmula, B refere-se à largura de banda do canal utilizado, L é o número de níveis de sinais utilizados para representação de dados e a C_N é a capacidade de transmissão de Nyquist, o número de *bits* por segundo, de um canal livre de ruído.

Exemplo 7

Considere o canal de voz com uma largura de banda de aproximadamente 3kHz, transmitindo um sinal codificado em dois níveis de tensão. A taxa máxima de transmissão de dados pode ser calculada como:

$$C_N = 2 \times B \times \log_2 L = 2 \times 3000 \times \log_2 2 = 6.000 \text{ bps}$$

Exemplo 8

Considere o mesmo canal sem ruídos, transmitindo um sinal codificado em quatro níveis (para cada nível são enviados 2 *bits* por vez). A taxa máxima de transmissão de dados pode ser calculada como:

$$C_N = 2 \times B \times \log_2 L = 2 \times 3000 \times \log_2 4 = 12.000 \text{ bps}$$

Canal com Ruído: Lei de Shannon

Na realidade, um canal sem ruído é uma idealização. Sempre haverá ruído num canal. Em 1944, Claude Shannon introduziu um resultado, que hoje leva o nome dele, para determinar o limite teórico máximo da transmissão de dados para um canal com ruído:

$$C_s = B \times \log_2 (1 + \text{SNR})$$

Outra vez, B refere-se à largura de banda do canal utilizado. SNR é a **razão sinal-ruído** do canal e a C_s é a capacidade de transmissão de Shannon, ou seja, o número de *bits* por segundo de um canal na presença de ruído.

A razão sinal-ruído é uma relação estatística entre a potência do sinal pela potência do ruído no canal. Note que a fórmula de Shannon não menciona o número de níveis do sinal. Isto significa que não importa quantos níveis usamos, não conseguiremos transmitir dados numa taxa superior à capacidade imposta pelo canal. Noutras palavras, a fórmula determina uma característica do canal, não o método de transmissão.

Exemplo 9

Considere um canal com um nível de ruído extremamente alto. Nesse caso, podemos aproximar a razão sinal-ruído do canal para zero. Uma razão sinal-ruído nula indica que, não importa os esforços realizados, é impossível transmitir um sinal de comunicação através desse canal, pois o nível de ruído mascara e destroi completamente a informação que é feita passar no meio. Para o canal supracitado a capacidade é calculada através de:

$$C_s = B \times \log_2 (1 + \text{SNR}) = B \times \log_2 (1+0) = B \times \log_2 (1) = 0$$

Isto significa que a capacidade de transmissão é zero independentemente da largura de banda desse canal. De fato, não podemos receber dados transmitidos através desse tipo canal.

Exemplo 10

Vamos calcular o limite teórico máximo para a transmissão de dados através do canal de voz tradicional. Uma linha telefônica normalmente possui uma largura de banda de 3000Hz (300Hz a 3300Hz). A razão sinal-ruído de uma linha telefônica boa vale 3162. Para este canal, a capacidade é calculada por:

$$C_s = B \times \log_2 (1 + \text{SNR}) = 3000 \times \log_2 (1+3162) = 3000 \times \log_2 (3163)$$

$$C_s = 3000 \times 11,62 = 34.860 \text{ bps}$$

Isto significa que a maior taxa de transmissão de dados através da linha telefônica é aproximadamente 34Kbps. Se quisermos transmitir dados em velocidades maiores que esta, podemos aumentar a largura de banda da linha ou melhorar (aumentar) a razão sinal-ruído.

Usando Ambos Limites

Na prática, precisamos utilizar ambos métodos para determinar que largura de banda e quantos níveis de codificação serão necessários numa transmissão. Vamos ilustrar isso através de um exemplo.

Exemplo 11

Suponha um canal com uma largura de banda de 1MHz e uma razão sinal-ruído de 63. Qual é a capacidade máxima desse canal e o número de níveis de codificação apropriados à transmissão?

Solução

Primeiramente, usamos a lei de Shannon para determinar o limite superior de transmissão do canal.

$$C_s = B \times \log_2 (1 + \text{SNR}) = 10^6 \times \log_2 (1 + 63) = 10^6 \times \log_2 (64) = 6 \text{ Mbps}$$

Embora a fórmula de Shannon dê 6Mbps, este é o limite superior. Para uma melhor performance escolhemos, arbitrariamente, um limite inferior: por exemplo 4Mbps. Em seguida, usamos a fórmula de Nyquist para determinar o número de níveis de codificação.

$$4 \text{ Mbps} = 2 \times 1 \text{ MHz} \times \log_2 L \Rightarrow L = 4$$

3.6 TRANSMISSÃO COM PERDAS

Os meios de transmissão por onde enviamos sinais não são perfeitos. As imperfeições do meio provocam o enfraquecimento e deformação do sinal. Isto significa que a potência e o padrão do sinal no transmissor e no receptor, acoplados nas duas extremidades do meio, não são as mesmas. Assim, o que é enviado não é aquilo que é recebido. Existem três tipos de perdas comuns em meios: atenuação, distorção e ruído (veja a Fig. 3.20).

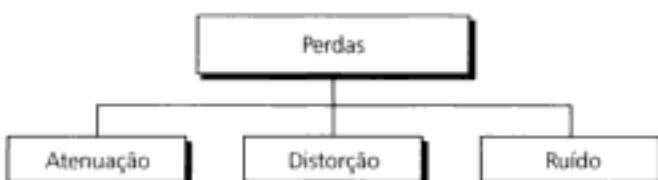


Figura 3.20 Tipos de perdas.

Atenuação

Atenuação significa perda de energia. Quando um sinal, simples ou composto, viaja num meio, irremediavelmente perde energia. Muitas vezes essa perda é associada à resistência do meio. No caso de meios metálicos que transportam corrente elétrica, um efeito de aquecimento (denominado efeito Joule) provoca atenuação do sinal. Uma parte da energia transportada pelo sinal no meio é convertida em calor e perdida durante o processo de transmissão. Para compensar essa perda, amplificadores podem ser utilizados para restaurar o nível do sinal*. A Figura 3.21 mostra os efeitos da atenuação e da amplificação.



Figura 3.21 Atenuação.

Decibel

Para lidar com sinais de diferentes níveis de potência, os engenheiros usam o conceito de decibel. O **decibel (dB)** mede as intensidades relativas entre dois sinais ou um mesmo sinal em dois pontos diferentes. Para manter as coisas em ordem, perceba que o decibel deve ser negativo se um sinal é atenuado e positivo se um sinal for amplificado.

$$\text{dB} = 10 \times \log_{10}(P_2/P_1)$$

onde P_1 e P_2 são as potências do sinal nos pontos 1 e 2, respectivamente.

* N. de R. T.: Um amplificador é excelente, mas não faz milagre. Para que o nível de potência de um sinal seja restaurado, o amplificador deve retirar energia de uma fonte CC externa. Assim, a lei de conservação da energia permanece intacta e inviolável.

Exemplo 12

Imagine que ao viajar através de um meio um sinal perca metade da potência original. Isto significa que $P_2 = 1/2P_1$. Nesse caso, a atenuação (perda de potência) pode ser calculada por:

$$10 \times \log_{10}(P_2/P_1) = 10 \times \log_{10}(0,5 P_1/P_1) = 10 \times \log_{10}(0,5) = 10 \times (-0,3) = -3\text{dB}$$

Técnicos e engenheiros sabem que -3dB ou uma perda de 3dB é equivalente a uma redução de metade da potência de um sinal.

Exemplo 13

Imagine que um sinal é amplificado 10 vezes por um amplificador. Isto significa que $P_2 = 10 \times P_1$. Nesse caso, o nível de amplificação de potência em decibel é calculado por:

$$10 \times \log_{10}(P_2/P_1) = 10 \times \log_{10}(10P_1/P_1) = 10 \times \log_{10}(10) = 10 \times (1) = 10\text{dB}$$

Exemplo 14

Uma das razões dos engenheiros utilizarem o decibel para medir as variações de intensidade de um sinal é que os números em decibel podem ser somados ou subtraídos diretamente, quando muitos pontos estiverem sendo analisados simultaneamente (análise em cascata dos pontos). Na Figura 3.22, um sinal viaja uma grande distância entre os pontos 1 e 4. O sinal é atenuado entre os pontos 1 e 2. Entre os pontos 2 e 3 o sinal é amplificado. Novamente, entre os pontos 3 e 4 o sinal é atenuado. Podemos determinar o nível em decibel resultante entre os pontos 1 e 4 adicionando as medidas em decibel entre esse conjunto de pontos.

Nesse caso, o nível em decibel total pode ser calculado por:

$$\text{dB} = -3 + 7 - 3 = +1$$

Este resultado indica que, ao viajar entre os pontos 1 e 4, o sinal aumentou o nível de potência.

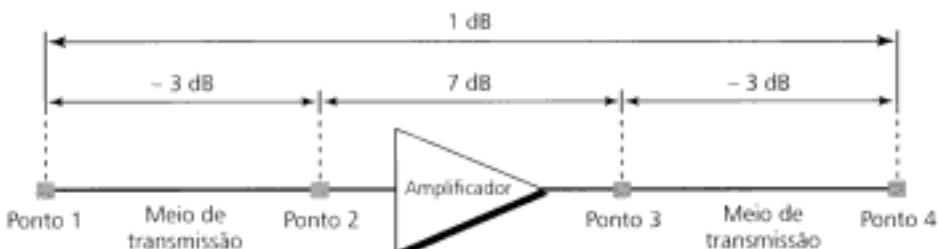


Figura 3.22 Exemplo 14.

Distorção

A **distorção** é alteração da forma de um sinal ao propagar-se num meio ou ao ser amplificado em um circuito. A distorção ocorre freqüentemente em sinais compostos. O problema é que cada componente do sinal possui uma velocidade de propagação (veja a próxima seção) característica através do meio de suporte e, portanto, atrasa de modo diferente para chegar ao destino final. A Figura 3.23 mostra o efeito da distorção em um sinal composto.

Ruído

O **ruído** é outro problema. Um sinal pode ser corrompido por diversos tipos de ruídos diferentes, tais como o ruído térmico, ruído induzido, *crosstalk* e o ruído impulsivo. O ruído térmico é provocado pelo movimento aleatório (agitação térmica) de elétrons nos condutores que gera um sinal extra, diferente daquele originado no transmissor. O ruído induzido é provocado pelo acionamento



Figura 3.23 Distorção.

de cargas indutivas, tais como de motores e outros aparelhos. Estas cargas agem como antenas transmissoras, enquanto que o meio de transmissão age como antena receptora. *Crosstalk* é o efeito que uma corrente num condutor provoca no outro. Um condutor age como antena transmissora e o outro como antena receptora. Por fim, o ruído impulsivo é uma resposta abrupta no meio (um sinal com uma energia muito alta por um intervalo de tempo muito curto) proveniente de redes elétricas, de iluminação e outras fontes. A Figura 3.24 ilustra o efeito de um ruído sobre um sinal viajando num meio.



Figura 3.24 Ruidos.

3.7 UM POUCO MAIS SOBRE SINAIS

Existem outros quatro parâmetros de medida usados na comunicação de dados: *throughput*, velocidade de propagação, tempo de propagação e comprimento de onda. Discutiremos estes parâmetros nesta seção, antes de fechar o capítulo.

Throughput

O *throughput* é uma medida da velocidade com que os dados cruzam um ponto ou uma rede. Noutras palavras, se considerarmos o ponto como sendo um plano vertical que secciona o meio, o *throughput* é o número de *bits* que atravessa esse plano em um segundo. A Figura 3.25 apresenta o conceito.

Velocidade de Propagação

A **velocidade de propagação** é uma medida da distância que um sinal ou *bit* pode viajar, através de um meio, numa unidade de tempo de 1 segundo. A velocidade de propagação de sinais eletromagnéticos depende do meio e da freqüência do sinal. Por exemplo, no vácuo, a luz propaga-se com uma velocidade de 3×10^8 m/s. A velocidade da luz no ar é menor que o valor do vácuo e é menor ainda dentro de uma fibra óptica.

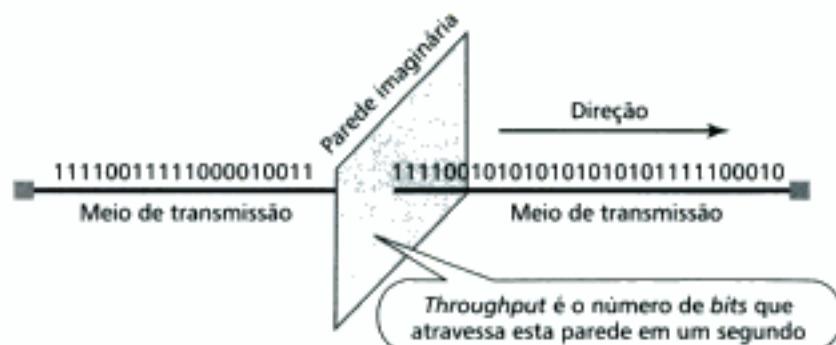


Figura 3.25: Throughput.

Tempo de Propagação

O **tempo de propagação** mede o tempo necessário para que um sinal ou um *bit* viaje de um ponto específico no meio de transmissão a outro. O tempo de propagação é calculado dividindo a distância percorrida pela velocidade de propagação do sinal. A Figura 3.26 ilustra o conceito.

$$\text{Tempo de propagação} = \text{distância}/\text{velocidade de propagação}$$

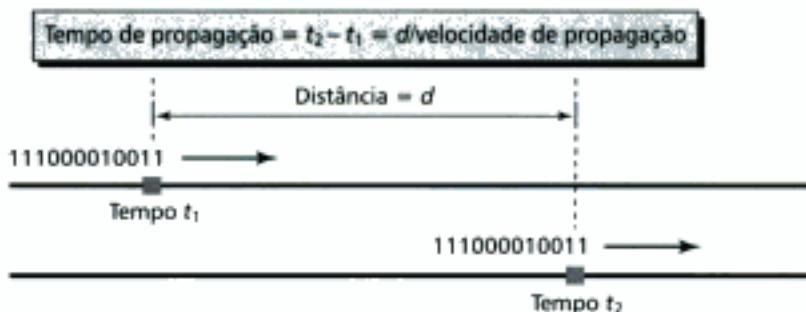


Figura 3.26: Tempo de propagação.

Comprimento de Onda

O **comprimento de onda** é outra característica importante de um sinal que viaja num meio de transmissão. O comprimento de onda une o período ou a freqüência de uma senóide simples com a velocidade de propagação do meio. De outra maneira, enquanto a freqüência de um sinal independe do meio, o comprimento de onda guarda uma relação estreita com a freqüência e o meio. Embora possamos associar um comprimento de onda a sinais elétricos, é melhor utilizá-lo quando estivermos lidando com a transmissão de luz numa fibra óptica ou com transmissão de ondas eletromagnéticas em meios abertos. O comprimento de onda é a distância que um sinal simples pode viajar durante um período do sinal (veja Figura 3.27).

Podemos determinar o comprimento de onda se forem conhecidos a velocidade de propagação e o período do sinal.

$$\text{Comprimento de onda} = \text{velocidade de propagação} \times \text{período}$$

Além disso, como período e freqüência estão relacionados entre si, também podemos fazer:

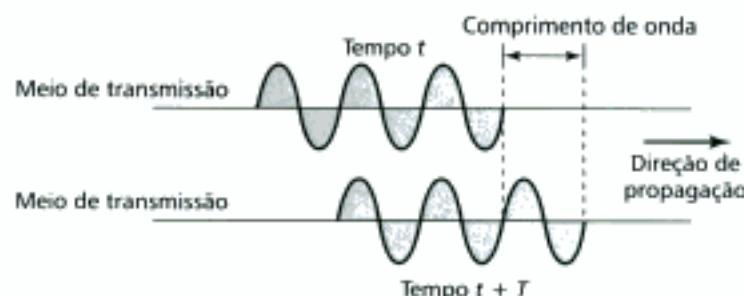


Figura 3.27 Comprimento de onda.

Comprimento de onda = Velocidade de propagação × (1/freqüência) = Velocidade de propagação/freqüência

Se representarmos o comprimento de onda através da letra grega λ (lê-se lambda), a velocidade de propagação por c (velocidade da luz) e a freqüência por f , temos:

$$\lambda = c/f$$

Em comunicação de dados, o comprimento de onda normalmente é medido em micrômetros (microns). Por exemplo, vamos determinar o comprimento de onda da luz vermelha ($f = 4 \times 10^{14} \text{ Hz}$) no ar:

$$\lambda = \frac{c}{f} = \frac{3 \times 10^8 \text{ m/s}}{4 \times 10^{14} \text{ Hz}} = \frac{3 \times 10^8 \text{ m/s}}{4 \times 10^{14} (1/\text{s})} = 0,75 \times 10^{-6} \text{ m} = 0,75 \mu\text{m}$$

Entretanto, num cabo coaxial ou numa fibra óptica o comprimento de onda é menor ($0,5 \mu\text{m}$) porque a velocidade de propagação dentro do cabo é menor que a velocidade de propagação no ar.

3.8 TERMOS-CHAVE

Amplitude de pico	Informação analógica
Análise de Fourier	Informação digital
Atenuação	Intervalo de sinalização
Bits por segundo (bps)	Largura de banda
Canal passa-baixas	Lei de Shannon
Canal passa-banda	Número de bits por segundo
Ciclo	Onda senoidal
Comprimento de onda	Período
Decibel (dB)	Razão sinal-ruído (SNR)
Digital	Ruído
Distorção	Sinal
Domínio da freqüência	Sinal analógico
Domínio do tempo	Sinal composto
Espectro de freqüência	Sinal digital
Fase	Sinal não periódico
Fórmula de Nyquist	Sinal periódico
Freqüência	Tempo de propagação
Freqüência fundamental	Throughput
Harmônicos	Velocidade de propagação
Hertz (Hz)	

3.9 RESUMO

- Dados devem ser convertidos em sinais eletrônicos antes de serem transmitidos através de uma rede.
- Dados e sinais podem ser tanto analógicos quanto digitais.
- Um sinal é periódico se ele consiste de um padrão repetido continuamente.
- Toda onda senoidal pode ser caracterizada pela amplitude, freqüência e fase.
- Freqüência e período são grandezas recíprocas entre si.
- Num gráfico no domínio do tempo plotamos a amplitude de um sinal em função do tempo.
- Num gráfico no domínio da freqüência plotamos as amplitudes de resposta (*spikes*) de um sinal composto em função das freqüências.
- Usando a análise de Fourier, qualquer sinal composto pode ser representado como uma combinação de ondas senoidais.
- O espectro de freqüência de um sinal consiste dos componentes de ondas senoidais necessários para a construção da forma de onda do sinal.
- A largura de banda de um sinal é a faixa de freqüências que o sinal ocupa. A largura de banda pode ser determinada tomando-se a diferença entre a maior e a menor componentes de freqüência.
- O número de *bits* por segundo (*bit rate*) e o intervalo de sinalização (tempo de duração de 1 *bit*) são termos utilizados na descrição de sinais digitais.
- Um sinal digital é um sinal composto com uma largura de banda infinita.
- O número de *bits* por segundo e a largura de banda são proporcionais entre si.
- A fórmula de Nyquist determina a taxa de transmissão teórica de um canal livre de ruídos.
- A lei de Shannon determina a capacidade máxima de transmissão de dados para um canal com ruído.
- Atenuação, distorção e ruído provocam perdas no sinal.
- Atenuação é a perda de energia do sinal devido às resistências do meio.
- O decibel mede as intensidades relativas entre dois sinais ou de um sinal em dois pontos diferentes.
- A distorção é a alteração de um sinal devido às diferentes velocidades de propagação das componentes de freqüência que compõem o sinal.
- Ruídos são fontes de perturbação externas que corrompem o sinal.
- Outras características dos meios de transmissão são o *throughput*, a velocidade e o tempo de propagação.
- O comprimento de onda de um sinal de freqüência qualquer é definido como o quociente da velocidade de propagação pela freqüência.

3.10 PRATIQUE OS CONHECIMENTOS ADQUIRIDOS

Questões de Revisão

1. Descreva as três características de uma onda senoidal.
2. O que é espectro de freqüência de um sinal?
3. Compare um sinal analógico com um sinal digital.
4. Um sinal detectado no receptor possui os seguintes níveis: -1, 0 e 1. Este sinal é analógico ou digital?
5. Qual é a relação entre período e freqüência?
6. Quais são as unidades de medida do período e da freqüência?
7. O que indica a amplitude de um sinal?
8. O que indica a freqüência de um sinal?
9. O que indica a fase de um sinal?
10. Em que tipo de gráfico mostramos a amplitude de um sinal em um dado instante de tempo?
11. Como um sinal composto pode ser decomposto nas componentes de freqüência que o compõem?
12. O que é intervalo de sinalização e qual é a contrapartida dessa grandeza num sinal analógico?
13. O que é número de *bits* por segundo e qual é a contrapartida dessa grandeza num sinal analógico?
14. Cite os três tipos de perdas em meios de transmissão.

15. O que o decibel mede?
16. Qual é a relação entre a velocidade e o tempo de propagação?
17. O que é comprimento de onda de um sinal e como podemos calculá-lo?

Questões de Múltipla Escolha

19. Antes de transmitir os dados, devemos transformá-los em _____.
 a. Sinais periódicos
 b. Sinais eletromagnéticos
 c. Sinais não periódicos
 d. Ondas senoidais de baixa freqüência
20. Um sinal periódico completa um ciclo em 0,001s. Qual é a freqüência desse sinal?
 a. 1Hz
 b. 100Hz
 c. 1kHz
 d. 1MHz
21. Qual das seguintes opções podemos determinar através de um gráfico no domínio da freqüência?
 a. Freqüência
 b. Fase
 c. Potência
 d. Todas as respostas acima
22. Qual das seguintes opções podemos determinar através de um gráfico no domínio da freqüência?
 a. Largura de banda
 b. Fase
 c. Potência
 d. Todas as respostas acima
23. Num gráfico no domínio da freqüência, o eixo vertical mede a _____.
 a. Amplitude de pico
 b. Freqüência
 c. Fase
 d. Inclinação
24. Num gráfico no domínio da freqüência, o eixo horizontal mede a _____.
 a. Amplitude de pico
 b. Freqüência
 c. Fase
 d. Inclinação
25. Num gráfico no domínio do tempo, o eixo vertical é uma medida da _____.
 a. Amplitude
 b. Freqüência
 c. Fase
 d. Tempo
18. O que a lei de capacidade de Shannon impõe às comunicações?
26. Num gráfico no domínio do tempo, o eixo horizontal é uma medida da _____.
 a. Amplitude do sinal
 b. Freqüência
 c. Fase
 d. Tempo
27. Se a largura de banda de um sinal vale 5kHz e a menor freqüência da faixa é 52kHz, qual é a maior freqüência?
 a. 5kHz
 b. 10kHz
 c. 47kHz
 d. 57kHz
28. Qual é a largura de banda de um sinal que cobre a faixa de 40kHz a 4MHz?
 a. 36MHz
 b. 360kHz
 c. 3,96MHz
 d. 396kHz
29. Quando uma das componentes de um sinal possui freqüência zero, a amplitude média do sinal _____.
 a. É maior que zero
 b. É menor que zero
 c. É igual a zero
 d. (a) ou (b)
30. Um sinal periódico sempre pode ser decomposto em _____.
 a. Um número exato de ondas senoidais de freqüências ímpares
 b. Um conjunto de ondas senoidais
 c. Um conjunto de ondas senoidais, uma das quais deve possuir fase 0°
 d. Nenhuma das respostas anteriores
31. Quando a freqüência aumenta, o período _____.
 a. Diminui
 b. Aumenta
 c. Permanece o mesmo
 d. Dobra
32. Dadas duas ondas senoidais *A* e *B*, se a freqüência de *A* é duas vezes maior que a freqüência de *B*, então o período de *B* é _____ período de *A*.
 a. Metade do
 b. Duas vezes o
 c. Idêntico ao
 d. Indeterminado a partir do

33. Uma onda senoidal é _____.
 a. Periódica e contínua
 b. Não periódica e contínua
 c. Periódica e discreta
 d. Não periódica e discreta
34. Se a amplitude de pico de uma senóide é 2V, a amplitude mínima vale _____.
 a. 2
 b. 1
 c. -2
 d. Entre -2 e 2
35. Um sinal é medido em dois pontos diferentes. A potência do sinal é P_1 , no primeiro ponto e P_2 no segundo. Sabendo que a relação entre os níveis em decibel vale 0, significa que _____.
 a. P_2 é zero.
 b. P_2 é igual a P_1 .
 c. P_2 é maior que P_1 .
 d. P_2 é menor que P_1 .
36. _____ é um tipo de perda de transmissão na qual o sinal perde a intensidade devido às resistências que o meio de transmissão impõe ao fluxo do sinal.
 a. Atenuação
 b. Distorção
 c. Ruído
 d. Decibel
37. _____ é um tipo de perda na transmissão na qual o sinal perde a intensidade devido às diferentes velocidades de propagação que cada componente de frequência possui no meio.
 a. Atenuação
 b. Distorção
 c. Ruído
 d. Decibel
38. _____ é um tipo de perda de transmissão na qual uma fonte externa, tal como um *crosstalk*, corrompe o sinal.
 a. Atenuação
 b. Distorção
 c. Ruído
 d. Decibel
39. _____ é medida em metros/segundo ou quilômetros/segundo.
 a. Throughput
 b. Velocidade de propagação
 c. Ruído
 d. Decibel
40. _____ é medido em bits/segundo.
 a. Throughput
- b. Velocidade de propagação
 c. Tempo de propagação
 d. (b) ou (c)
41. _____ é medido em segundos.
 a. Throughput
 b. Velocidade de propagação
 c. Tempo de propagação
 d. (b) ou (c)
42. Quando multiplicamos a velocidade pelo tempo de propagação, obtemos _____.
 a. Throughput
 b. Comprimento de onda do sinal
 c. Fator de distorção
 d. Distância que um sinal ou um bit viaja num meio
43. O tempo de propagação é _____ proporcional à distância e _____ proporcional à velocidade de propagação.
 a. Inversamente; diretamente
 b. Diretamente; inversamente
 c. Inversamente; inversamente
 d. Diretamente; diretamente
44. O comprimento de onda é _____ proporcional à velocidade de propagação e _____ proporcional ao período.
 a. Inversamente; diretamente
 b. Diretamente; inversamente
 c. Inversamente; inversamente
 d. Diretamente; diretamente
45. O comprimento de onda de um sinal depende do(a) _____.
 a. Freqüência do sinal
 b. Meio
 c. Fase do sinal
 d. (a) e (b)
46. O comprimento de onda da luz verde no ar é _____ o comprimento de onda da luz verde dentro de uma fibra óptica.
 a. Menor que
 b. Maior que
 c. Igual ao
 d. Nenhuma das opções acima
47. Usando a fórmula de Shannon para determinar a taxa máxima de transmissão de dados para um certo canal, se $C = B$, então _____.
 a. O sinal é menor que o ruído
 b. O sinal é maior que o ruído
 c. O sinal é igual ao ruído
 d. Não há informação suficiente para responder essa questão

Exercícios

48. Considerando as freqüências listadas abaixo, determine os períodos correspondentes. Expresse suas respostas em segundos, milissegundos, microssegundos, nanossegundos e picossegundos.
- 24Hz
 - 8MHz
 - 140kHz
 - 12THz
49. Considerando os períodos listados abaixo, determine as freqüências correspondentes. Expresse suas respostas em hertz, kilohertz, megahertz, gigahertz e terahertz.
- 5s
 - $12\mu s$
 - 220ns
 - 81ps
50. Qual é o deslocamento de fase nos seguintes casos?
- Uma onda senoidal com a amplitude de pico posicionada no tempo zero.
 - Uma onda senoidal com a amplitude de pico posicionada no ponto $1/4$ de ciclo
 - Uma onda senoidal com valor instantâneo zero posicionado no ponto $3/4$ de ciclo
 - Uma onda senoidal com a amplitude mínima posicionada no ponto de $1/4$ de ciclo
51. Apresente o deslocamento de fase, em graus, correspondente a cada uma dos seguintes atrasos em relação a um ciclo.
- 1 ciclo
 - $1/2$ ciclo
 - $3/4$ ciclo
 - $1/3$ ciclo
52. Apresente os atrasos, em ciclo, correspondentes a cada um dos seguintes deslocamentos de fase, em graus:
- 45°
 - 90°
 - 60°
 - 360°
53. Desenhe uma senóide no domínio do tempo, num intervalo de 1s, com amplitude de pico de 15V, uma freqüência 5Hz e uma fase de 270° .
54. Desenhe duas senóides no mesmo gráfico no domínio do tempo. As características das senóides são as seguintes:
- Sinal A: amplitude = 40V, freqüência = 9Hz, fase = 0° .
 Sinal B: amplitude = 10V, freqüência = 9Hz, fase = 90° .
55. Desenhe dois períodos de uma onda senoidal, deslocada em fase 90° . Em seguida, desenhe outra senóide, no mesmo gráfico no domínio do tempo, de mesma amplitude e freqüência, mas com uma fase de 90° em relação à primeira senóide.
56. Qual é a largura de banda de um sinal passível de ser decomposto em quatro senóides de freqüências 0, 20, 50 e 200Hz? Se todos os harmônicos têm a mesma amplitude, desenhe o espectro de freqüência desse sinal.
57. Um sinal composto e periódico, possuindo uma largura de banda de 2kHz, é decomposto em dois harmônicos. O primeiro tem uma freqüência de 100Hz e uma amplitude de 20V. O segundo tem amplitude de 5V. Desenhe o espectro de freqüência desse sinal.
58. Mostre como um sinal senoidal pode ser modificado em fase desenhando dois períodos de uma onda senoidal arbitrária, com um deslocamento de fase de 0° , seguida de dois períodos do *mesmo sinal* com um deslocamento de fase de 90° .
59. Suponha que dispomos de um sinal senoidal A. Mostre o sinal negativo de A. Noutras palavras, mostre o sinal $-A$. Podemos relacionar o negativo de um sinal através de um deslocamento de fase? Caso a resposta seja afirmativa, qual é o deslocamento em graus?
60. Qual dos sinais compostos possui maior largura de banda, um sinal que varia 100 vezes por segundo ou um sinal que varia 200 vezes por segundo? Justifique sua resposta.
61. Qual é o número de bits por segundo dos seguintes sinais?
- Um sinal tal que 1 bit dura 0,001s
 - Um sinal tal que 1 bit dura 2ms
 - Um sinal tal que 10 bits duram 20μs
 - Um sinal tal que 1000 bits duram 250ps
62. Qual é a duração de 1 bit para cada um dos seguintes sinais?
- Um sinal tal que o número de bits por segundo vale 100bps.
 - Um sinal tal que o número de bits por segundo vale 200kbps.

- c. Um sinal tal que o número de *bits* por segundo vale 5Mbps.
 d. Um sinal tal que o número de *bits* por segundo vale 1Gbps.
63. Um dispositivo está enviando dados a uma taxa de 1000bps.
 a. Quanto tempo leva para enviar 10 *bits*?
 b. Quanto tempo leva para enviar um único caractere (8 *bits*)?
 c. Quanto tempo leva para enviar um arquivo com 100.000 caracteres?
64. Qual é o número de *bits* por segundo do sinal mostrado na Figura 3.28?
65. Qual é a freqüência do sinal mostrado na Figura 3.29?
66. Desenhe a representação do sinal no domínio do tempo (nos primeiros 1/100s) correspondente à resposta no domínio da freqüência mostrada na Figura 3.30.
67. Desenhe a representação no domínio da freqüência do sinal mostrado na Figura 3.31.
68. Qual é a largura de banda do sinal mostrado na Figura 3.32?
69. Qual é a largura de banda do sinal mostrado na Figura 3.33?
70. Um sinal composto tem freqüências igualmente distribuídas entre 10 e 30kHz, todos com a mesma amplitude de 10V? Desenhe o espectro de freqüência.
71. Um sinal composto possui freqüências distribuídas numa faixa entre 10 e 30kHz. A amplitude do sinal de menor freqüência vale zero e do maior sinal, o que possui freqüência de 20kHz, vale 30V. Considerando que as amplitudes mudam gradualmente do mínimo até o máximo, desenhe o espectro de freqüência.
72. Dois sinais possuem a mesma freqüência. Entretanto, sempre que o primeiro sinal está na amplitude máxima, o segundo sinal está no valor instantâneo zero. Qual é o deslocamento de fase entre os dois sinais?
73. Qual é a representação matemática de um sinal com uma amplitude de 10V, uma freqüência de 2,5kHz e uma fase de 30°?
74. Mostre o espectro de freqüência do seguinte sinal composto:
- $$s(t) = 8 + 3 \operatorname{sen} 100\pi t + 5 \operatorname{sen} 200\pi t$$
75. Qual é o período do sinal abaixo?
- $$s(t) = 4 \operatorname{sen} 628t$$
76. Uma co-senóide é uma onda senoidal deslocada 90° em fase. Mostre a representação matemática de uma onda senoidal do seguinte sinal co-senoidal.
- $$s(t) = \cos(2\pi f t + \pi)$$
77. Um certo canal de TV possui uma largura de banda de 6MHz. Se enviarmos um sinal digital utilizando esse canal, qual é a taxa de transmissão de dados se utilizarmos: um harmônico, três harmônicos e cinco harmônicos?
78. Um sinal viaja de um ponto *A* até um ponto *B*. No ponto *A*, o sinal tem um nível de potência de 100W. No ponto *B*, o nível de potência do sinal vale 90W. Qual é o valor da atenuação em decibel?
79. A atenuação de um sinal vale -10dB. Qual é o nível de potência do sinal de saída se o sinal de entrada (original) possui 5W?
80. Um sinal deve passar através de três amplificadores cascateados, cada qual com um ganho de 4dB. Qual é o ganho total do conjunto? Quantas vezes o nível de sinal que sai dos amplificadores é maior que o nível de sinal que entra?
81. Se o throughput entre um dispositivo e um meio de transmissão é 5kbps, quanto tempo leva para enviar 100.000bps através desse dispositivo?
82. Se a luz viaja aproximadamente 8 minutos do Sol até a Terra, qual é a distância aproximada entre esses dois corpos celestes?
83. Um sinal possui um comprimento de onda de 1μm no ar. Qual é a distância que a frente de onda viaja durante 5 períodos de tempo?
84. Uma linha de transmissão de dados possui uma razão sinal-ruído da ordem de 1000 e uma largura de banda de 4000kHz. Qual é a capacidade de transmissão de dados máxima que essa linha pode suportar?
85. Medimos a performance do canal de voz (largura de banda de 4 kHz). Quando o sinal no canal vale 10V, o ruído é 5mV. Qual é a capacidade máxima de transmissão de dados nessa linha? Se o limite inferior admissível de transmissão de dados for 28,8kbps, em quantos níveis precisaremos codificar o sinal?

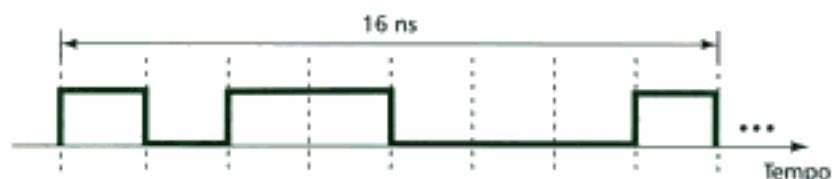


Figura 3.28 Exercício 64.

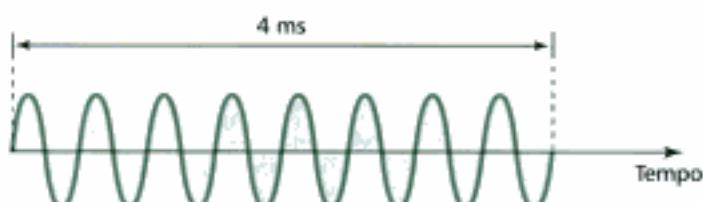


Figura 3.29 Exercício 65.



Figura 3.30 Exercício 66.

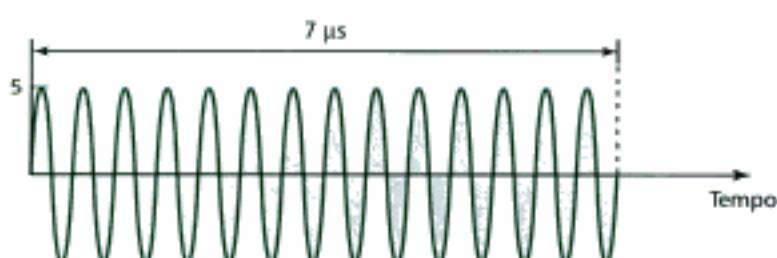


Figura 3.31 Exercício 67.

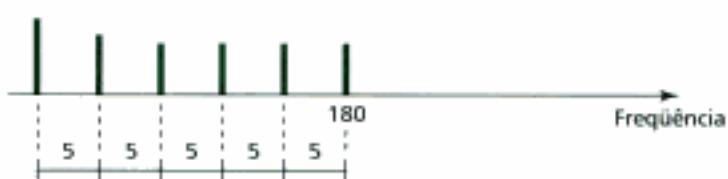


Figura 3.32 Exercício 68.



Figura 3.33 Exercício 69.

Capítulo 4

Transmissão Digital

Uma rede de computadores é projetada para enviar informação entre os dispositivos que a compõem. Durante a implementação de uma rede temos duas escolhas: converter a informação em sinal digital ou sinal analógico. Neste capítulo, discutiremos a primeira escolha: converter o sinal de transmissão para a forma digital. No Capítulo 5, discutiremos a segunda escolha, utilizando sinais analógicos.

No Capítulo 3, estudamos as vantagens e desvantagens da transmissão digital sobre a transmissão analógica. Neste capítulo, mostramos os esquemas e técnicas utilizadas para transmitir dados digitalmente. Em primeiro lugar, trataremos a codificação de linha, que é uma técnica para converter dados binários em sinais digitais. Em seguida, mostraremos como melhorar a eficiência da codificação de linha. Então, apresentaremos os conceitos e técnicas da amostragem de sinais que, em síntese, são técnicas utilizadas para converter informação analógica em dados binários. Após assumirem a forma binária, os dados podem ser colocados na forma de sinal digital através da técnica de codificação de linha, ou então, via uma combinação de codificação em bloco e codificação de linha. Finalmente, discutiremos as formas de transmissão serial e paralela de sinais digitais.

4.1 CODIFICAÇÃO DE LINHA

A **codificação de linha** é o processo de converter dados binários, ou seja, uma seqüência de *bits*, em sinais digitais. Por exemplo, dados, textos, números, imagens gráficas, áudio e vídeo que estão armazenados na memória do computador são todos seqüências de *bits* (veja Capítulo 1). A codificação de linha converte uma seqüência de *bits* em sinais digitais. A Figura 4.1 ilustra o conceito de codificação de linha.

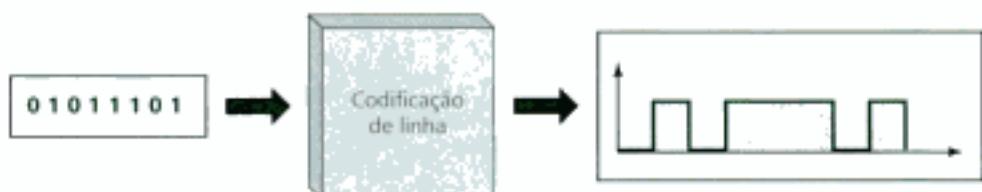


Figura 4.1 Codificação de linha.

Algumas Características da Codificação de Linha

Antes de discutirmos os diferentes tipos de codificação de linha, precisamos compreender as seguintes características fundamentais desse processo: níveis de codificação do sinal *versus* nível de dados, relógio (*clock*) *versus* número de *bits* por segundo, componentes CC, auto-sincronização e sincronização de relógios.

Nível de Sinal *versus* Nível de Codificação de Dados

Como foi discutido anteriormente, um sinal digital possui apenas um número limitado (finito) de valores. Entretanto, somente parte desses valores podem ser utilizados para representar dados, o restante é utilizado para outros propósitos, como veremos brevemente. Referimos à quantidade de valores permitidos num sinal como número de **níveis de sinal**. Ao número de valores utilizados para representar os dados damos o nome de **níveis de codificação de dados**. A Figura 4.2 mostra dois exemplos de sinais digitais. O primeiro sinal possui apenas dois níveis de sinal e dois níveis de codificação de dados. No segundo sinal temos três níveis de sinal e dois níveis de codificação de dados.

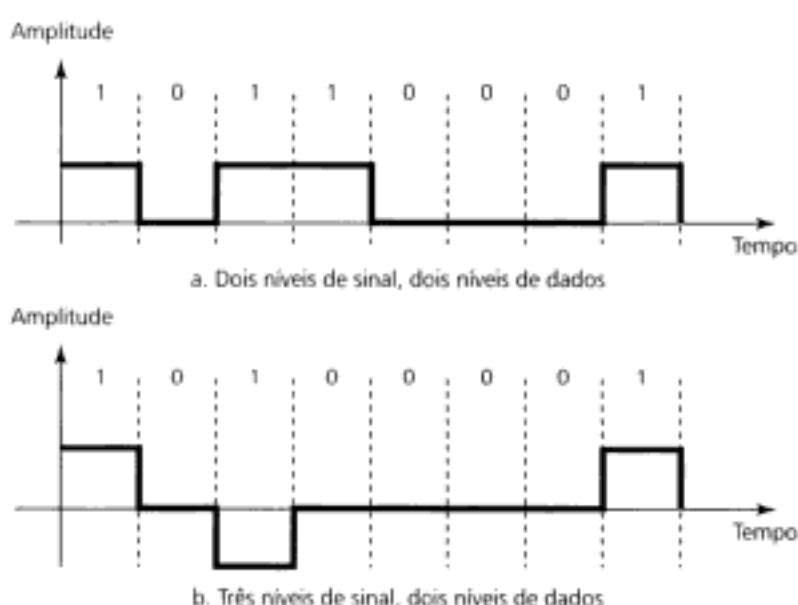


Figura 4.2 Nível de sinal *versus* nível de codificação de dados.

Relógio de Sincronismo *versus* Número de Bits por Segundo

O **relógio de sincronismo** (*clock*) define o número de pulsos por segundo. Um pulso é a quantidade de tempo mínima requerida para transmitir um símbolo. No Capítulo 3, vimos que o **número de bits por segundo** é a quantidade de *bits* enviados num intervalo de tempo de 1s. Se num pulso é transportado apenas 1 *bit*, o número de *bits* por segundo é equivalente ao número de pulsos por segundo, ou seja, ao relógio de sincronismo. Se a um pulso estão associados mais de 1 *bit*, então o número de *bits* por segundo é maior que o número de pulsos por segundo (relógio). Em geral, se L é o número de níveis de codificação de dados de um sinal, temos a seguinte expressão para o número de *bits* por segundo:

$$\text{Número de bits por segundo} = \text{Número de pulsos por segundo} \times \log L$$

Exemplo 1

Um sinal possui dois níveis de codificação de dados, com 1ms de duração de pulso. Vamos determinar o número de pulsos e de *bits* por segundo.

$$\text{Número de pulsos por segundo} = \frac{1}{1 \times 10^{-3}\text{s}} = 1000 \text{ pulsos/s}$$

$$\begin{aligned}\text{Número de bits por segundo} &= \text{Número de pulsos por segundo} \times \log_2 L \\ &= 1000 \times \log_2 2 = 1000 \text{ bps}\end{aligned}$$

Exemplo 2

Um sinal possui quatro níveis de codificação de dados com 1ms de duração de pulso. Vamos determinar o número de pulsos e de bits por segundo.

$$\text{Número de pulsos por segundo} = \frac{1}{1 \times 10^{-3}\text{s}} = 1000 \text{ pulsos/s}$$

$$\begin{aligned}\text{Número de bits por segundo} &= \text{Número de pulsos por segundo} \times \log_2 L \\ &= 1000 \times \log_2 4 = 2000 \text{ bps}\end{aligned}$$

Componentes DC

Alguns esquemas de codificação de linha não eliminam a componente DC de corrente contínua (freqüência zero) residual na linha. A componente DC é indesejável por duas razões. Primeiro, se este sinal tiver que passar através de um sistema que não permite a passagem da componente DC (tal como um transformador*), o sinal é distorcido e pode criar erros na saída. Segundo, a componente DC é energia extra inútil residente na linha. A Figura 4.3 mostra dois esquemas de codificação de linha. No primeiro há uma componente DC, as tensões positivas não são canceladas por tensões negativas. No segundo esquema não há nenhuma componente DC residual, as tensões positivas são canceladas pelas tensões negativas. O primeiro esquema de codificação não passa adequadamente através de um transformador, já o segundo não encontra problemas em passar.

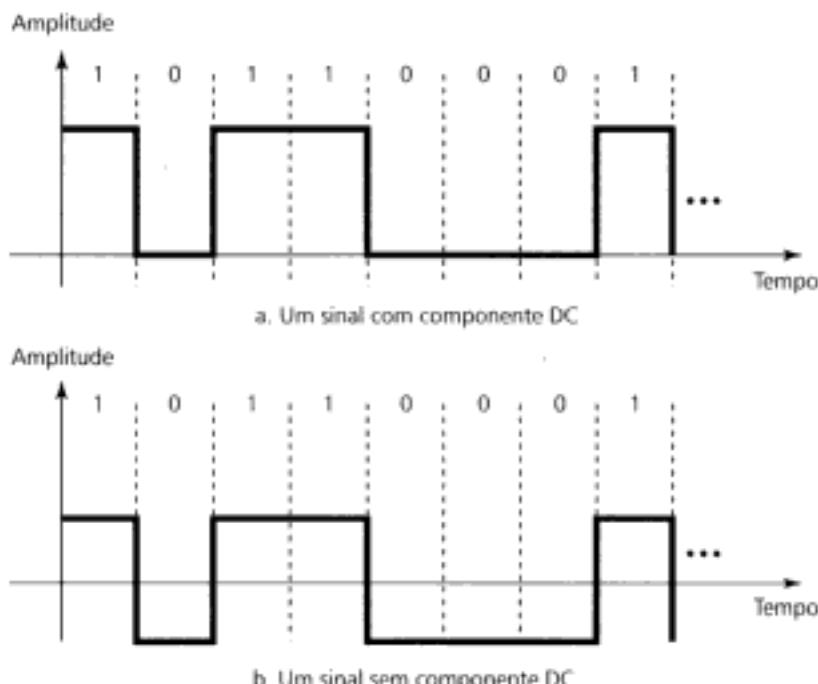


Figura 4.3 Componente DC.

Auto-sincronização

Para que os sinais oriundos do transmissor sejam interpretados corretamente no receptor, os intervalos de sinalização no receptor devem corresponder exatamente aos intervalos gerados no trans-

* N. de R. T.: Ainda hoje existem alguns esquemas de casamento de impedância, denominados pupinização, que utilizam transformadores de linha.

missor. Se a cadência do relógio (*clock*) do receptor for mais rápida ou mais lenta que o relógio do transmissor, os intervalos de sinalização não são encontrados e o receptor pode interpretar a comunicação diferentemente do que era pretendido pelo transmissor. A Figura 4.4 apresenta uma situação na qual o receptor possui um *bit* de curta duração. O transmissor envia 10110001, enquanto no receptor é entendido 110111000011 (situação exagerada).

Um sinal digital **auto-sincronizado** inclui a informação de sincronismo nos dados que estão sendo transmitidos. Geralmente, isto é feito introduzindo transições no sinal para alertar o receptor do começo, meio e fim do pulso. Se o relógio do receptor estiver fora de sincronismo, estas transições de alerta restauram (*reset*) o relógio do receptor.

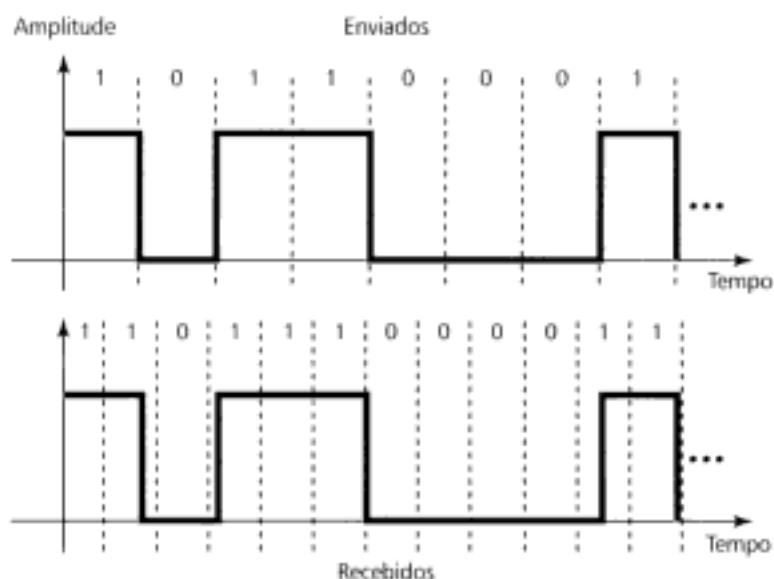


Figura 4.4 Auto-sincronização.

Exemplo 3

Numa transmissão digital, o relógio do receptor está 0,1% mais rápido que o relógio do transmissor. Quantos bits extras por segundo o receptor irá receber se a comunicação acontece numa taxa de 1 kbps? E a 1 Mbps?

Solução

Em 1 Kbps, o receptor receberá 1000 bps, em vez de 1000 bps.

1000 bits enviados \Rightarrow 1001 bits recibidos \Rightarrow 1 bit extra

Em 1Mbps, o receptor receberá 1.001.000 bps, em vez de 1.000.000 bps.

1.000.000 bits enviados \rightarrow 1.001.000 bits recibidos \rightarrow 1000 bits extras

Esquemas de Codificação

Conforme indicado na Figura 4.5, podemos dividir os esquemas de codificação em três grandes categorias: *unipolar*, *polar* e *bipolar*.



Figura 4.5 Esquemas de codificação.

Unipolar

O método de **codificação unipolar** é muito simples e muito primitivo. Embora esteja quase esquecido hoje em dia, esse é o método introdutório mais eficaz, no que tange ao desenvolvimento de conceitos, e que nos permite examinar os diversos tipos de problemas que os sistemas de codificação mais complexos devem ser capazes de suportar.

Os sistemas de transmissão digital trabalham enviando pulso de tensão ao longo do meio (*link*) que tipicamente são fios ou cabos. Em muitos métodos de codificação, um nível de tensão representa o nível binário 0 e outro nível representa o nível binário 1. A polaridade de um pulso diz se ele é positivo ou negativo. A codificação unipolar recebeu esse nome porque utiliza uma polaridade apenas. O sinal da polaridade pode ser atribuído a qualquer um dos dois estados binários, mas geralmente é deixado para o nível 1. Nesse caso, o outro estado (o nível 0) é representado por um zero de tensão.

A codificação unipolar utiliza somente um nível de tensão.

A Figura 4.6 mostra a idéia central da codificação unipolar. Nesse exemplo, os níveis UM são representados por uma tensão positiva e os níveis ZERO são codificados através do zero volts. De maneira objetiva, a codificação unipolar é barata e fácil de ser implementada.

Entretanto, a codificação unipolar possui pelo menos dois problemas sérios que a tornam indesejável: a componente DC residual e os problemas de sincronização. O valor médio do sinal codificado de modo unipolar fica claramente deslocado do referencial de OV. Isto origina uma componente DC na linha. O problema de sincronização é outro gargalo desse método de codificação. Se a seqüência de dados a serem representados contém longas filas de 0s ou de 1s, não há transições no sinal durante esse período para que o receptor possa ser alertado sobre possíveis problemas de sincronização.

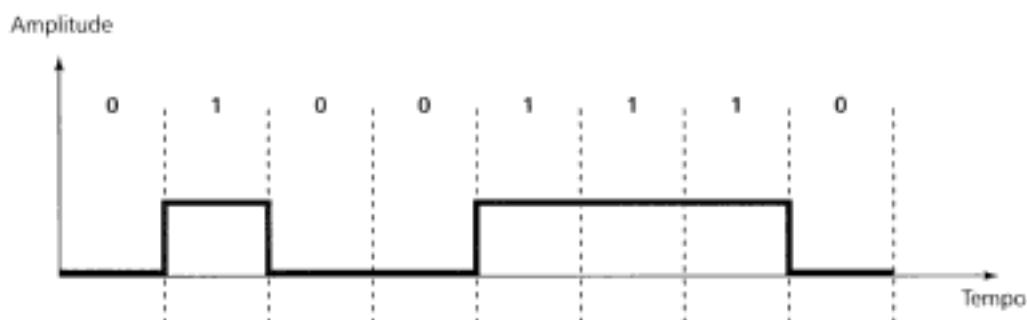


Figura 4.6 Codificação unipolar.

Polar

A **codificação polar** utiliza dois níveis de tensão, um positivo e outro negativo, para representar os dados. Assim, através de dois níveis de tensão a maioria dos métodos de codificação polar torna possível resolver o problema imediato do nível DC na linha encontrado na codificação anterior.

A codificação polar utiliza dois níveis de tensão (positivo e negativo).

Existem muitas variações de esquema de codificação polar. Examinaremos as quatro mais populares: **Non-Return to Zero (NRZ)**, **Return to Zero (RZ)**, **Manchester** e **Manchester Diferencial** (veja Figura 4.7).

Non-Return to Zero (NRZ) Na codificação NRZ, o valor do sinal é sempre positivo ou negativo. Existem duas formas conhecidas codificações NRZ.

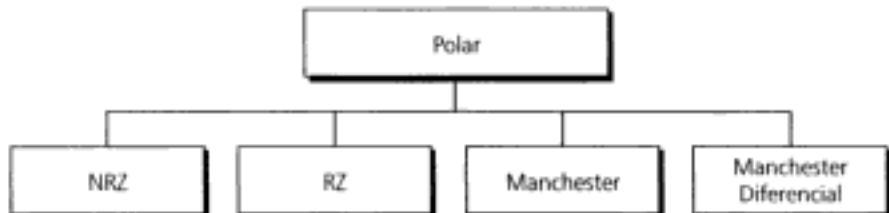


Figura 4.7 Tipos de codificação polar.

No esquema de codificação **NRZ-L** (NRZ-Level) o nível do sinal depende do *bit* que ele deve representar. Uma tensão positiva geralmente representa o *bit* 0, logo uma tensão negativa representa o *bit* 1. Assim, o nível do sinal está ligado ao estado do *bit* a ser representado. Isso pode dar origem a um outro problema, pois se os dados transmitidos contiverem uma cadeia longa de 0s ou 1s, a entrada do receptor será praticamente uma tensão contínua ao longo da sequência de 0s ou 1s. Desse modo, o receptor deve determinar quantos *bits* estão sendo enviados confiando unicamente no relógio que ele possui, o qual pode ou não estar sincronizado com o relógio do transmissor.

No esquema NRZ-L o nível do sinal depende do estado do bit.

Outra possibilidade é a **NRZ-I** (NRZ-Invert), onde qualquer transição entre níveis de tensão representa um *bit* 1. Estas transições devem ocorrer entre os níveis de tensão positivo e negativo para representar um *bit* 1. Assim, não é o nível de tensão em si que é utilizado para representar um *bit*, mas transições entre níveis representam o *bit* 1. Nessa codificação um *bit* 0 é representado pela ausência de transição.

O esquema NRZ-I é superior ao NRZ-L porque sempre que houver um *bit* 1 haverá uma transição de nível funcionando como mecanismo de sincronização. A existência de 1s na cadeia de dados permite ao receptor ajustar e sincronizar o relógio com o transmissor. Uma cadeia de 0s ainda pode causar problemas, mas sequências inteiras 0s não são assim tão prováveis na comunicação de dados, o que constitui um problema menor.

A Figura 4.8 ilustra as representações NRZ-L e NRZ-I da mesma sequência de *bits*. Na codificação NRZ-L, tensões positivas e negativas têm significados específicos: positivo para o *bit* 0 e negativo para o *bit* 1. Na codificação NRZ-I, os níveis de tensão em si não têm significado

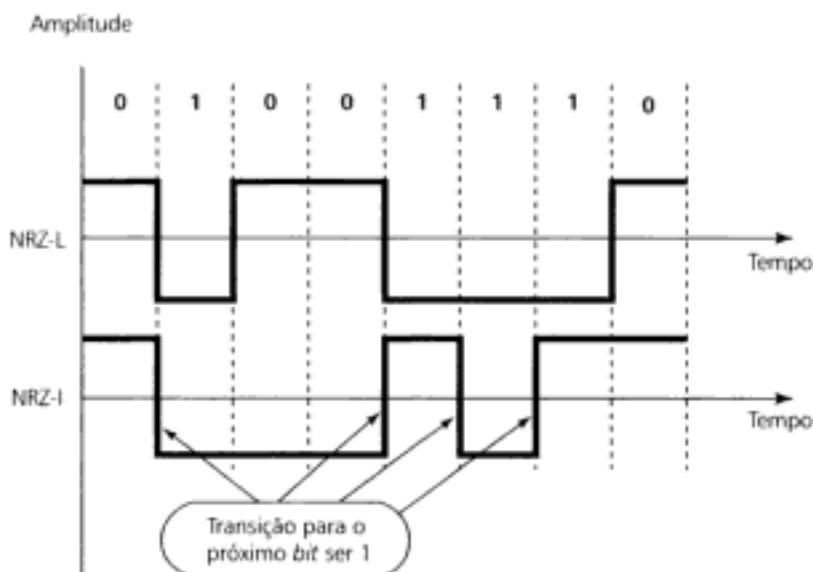


Figura 4.8 Codificação NRZ-L e NRZ-I.

isolado. Em vez disso, o receptor olha as transições entre níveis reconhecendo os *bits* em 1. Se durante um período de tempo não existirem transições o receptor encara o sinal como uma sequência de 0s.

Return to Zero (RZ) Como você pôde ver, numa transmissão de dados, a qualquer instante, cadeias inteiras e consecutivas de 0s ou 1s podem provocar perdas de dados no receptor. Uma solução para o problema é incluir o relógio de sincronização no sinal codificado, algo parecido com a solução proposta pelo esquema NRZ-I, mas também deve ser capaz de manipular sequências de 0s tão bem quanto as sequências de 1s.

Para garantir a sincronização no receptor, um bom esquema de codificação deve basear-se nas transições do sinal para cada *bit* representado. Então, o receptor pode usar essas transições para construir, atualizar e/ou sincronizar o relógio dele. Como vimos acima, no esquema NRZ-I foi resolvido o problema para as sequências de 1s. Se desejarmos algo parecido para representar os *bits* 0s de uma sequência, a solução mais simples é utilizar três níveis de tensão. O esquema RZ usa três valores de tensão: positivo, negativo e zero. Na codificação RZ, as transições do sinal não acontecem entre *bits*, mas durante cada *bit*. Como na solução NRZ-L, um nível positivo representa o *bit* 0 e um nível negativo representa o *bit* 1. Mas, diferentemente do esquema NRZ-L, na metade de cada intervalo de sincronização o sinal retorna a zero. Desse modo, um *bit* 1 é representado pela transição positivo-zero e um *bit* 0 é representado pela transição negativo-zero, não mais isoladamente através dos níveis de tensão. A Figura 4.9 ilustra esse conceito.

A maior desvantagem do esquema de codificação RZ é que ele requer duas transições de sinal para codificar um *bit* (0 ou 1) e, assim, ocupa uma largura de banda maior. Entretanto, das três alternativas analisadas até o momento essa é a mais eficaz.

Um bom esquema de codificação do sinal digital incorpora um relógio de sincronismo para o receptor.

Manchester A codificação Manchester usa uma inversão no meio de cada intervalo de sincronização tanto para a sincronização quanto para a representação do *bit*. Uma transição positiva (do nível de tensão negativo para o nível de tensão positivo) representa um *bit* 1 e uma transição negativa (do nível de tensão positivo para o nível de tensão negativo) representa um *bit* 0. Utilizando uma única tensão não-nula para os dois propósitos, a codificação Manchester exibe o mesmo nível de sincronização que a codificação RZ, mas usando somente dois níveis de tensão. A Figura 4.10 mostra um exemplo de codificação Manchester de uma sequência de *bits*.

Manchester Diferencial Na codificação Manchester Diferencial, a inversão no meio do intervalo é utilizada para sincronização, mas a presença ou ausência de uma transição adicional no começo do intervalo é usada para identificar o *bit*. Uma transição representa o *bit* 0 e a ausência de transição representa o *bit* 1. O esquema de codificação Manchester Diferencial requer duas transições no sinal para representar o *bit* 0, mas somente uma transição para representar o *bit* 1.

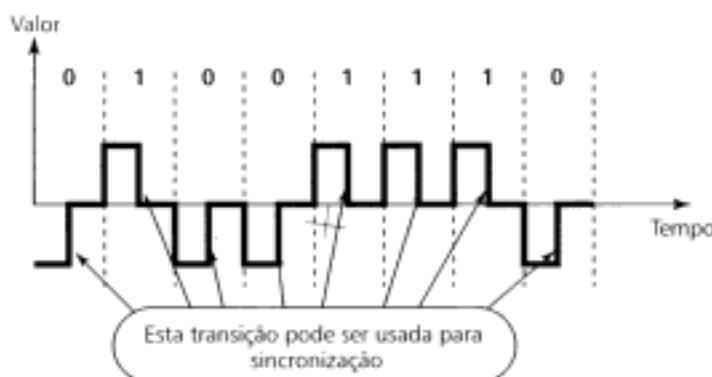


Figura 4.9 Codificação RZ.

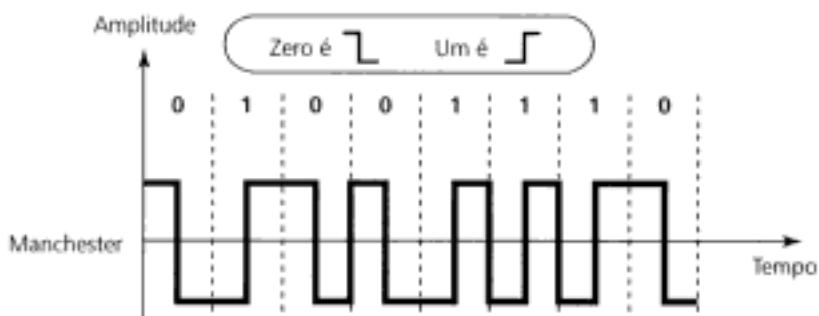


Figura 4.10 Codificação Manchester.

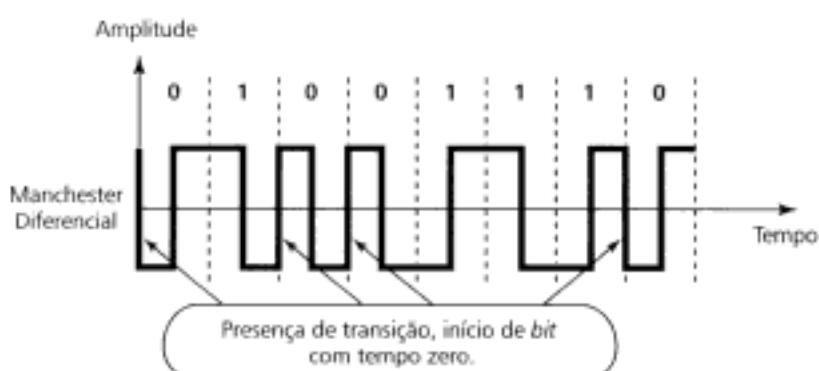


Figura 4.11 Codificação Manchester Diferencial.

No esquema de codificação Manchester Diferencial, a transição no meio do intervalo de um bit é utilizada somente como mecanismo de sincronização. A representação do bit é definida através de uma inversão ou não no início do bit.

Bipolar

Uma codificação bipolar, como a RZ, utiliza três níveis de tensão: positivo, negativo e zero. Entretanto, diferentemente da RZ, o nível zero na codificação bipolar é utilizada para representar um bit 0. Os 1s são representados através de pulsos alternados de tensão positiva e negativa. Se um primeiro bit 1 é representado através de uma tensão positiva, um segundo bit 1 será representado por uma tensão negativa, um terceiro bit 1 novamente através de uma tensão positiva e assim sucessivamente. Esta alternância ocorre até mesmo quando os bits 1 não são consecutivos.

Na codificação bipolar, usamos três níveis de tensão: positivo, negativo e zero.

Um esquema de codificação bipolar bastante difundido é denominado **Alternate Mark Inversion (AMI)**. Na expressão *alternate mark inversion*, a palavra *mark* (marca) foi tomada emprestada da telegrafia e significa 1. Então, AMI significa emitir um pulso sempre que um bit 1 for transmitido, de polaridade invertida em relação ao anterior. Uma tensão zero representa o bit 0. Finalmente, os bits 1s são representados através de pulsos de tensão alternados (positivos e negativos). A Figura 4.12 traz um exemplo de codificação AMI.

Uma modificação no esquema bipolar AMI foi desenvolvido para resolver o problema da sincronização seqüencial de 0s, especialmente nas transmissões de longas distâncias. Ele é denominado **BnZS (Bipolar n-Zero Substitution)**. Neste esquema, sempre que ocorrem *n* zeros consecutivos na seqüência, alguns dos bits nesse intervalo de *n* bits tornar-seão positivos ou negativos, o que agiliza a sincronização. Esta substituição viola as regras do AMI de um modo tão específico que o receptor sabe que estes bits são realmente 0s e não 1s.

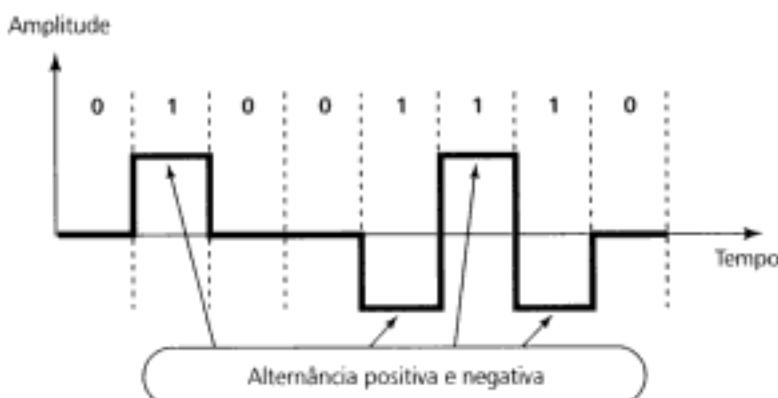


Figura 4.12 Codificação bipolar AMI.

Outros Esquemas

Existem outros esquemas de codificação de linha criados para finalidades específicas nas comunicações de dados. Discutiremos dois esquemas interessantes aqui: 2B1Q e MLT-3.

2B1Q

O 2B1Q (2 Binários, 1 Quaternário) utiliza quatro níveis de tensão, de tal modo que cada pulso é capaz de representar dois *bites* por vez, tornando-o assim mais eficiente. A Figura 4.13 mostra um exemplo de um sinal 2B1Q.

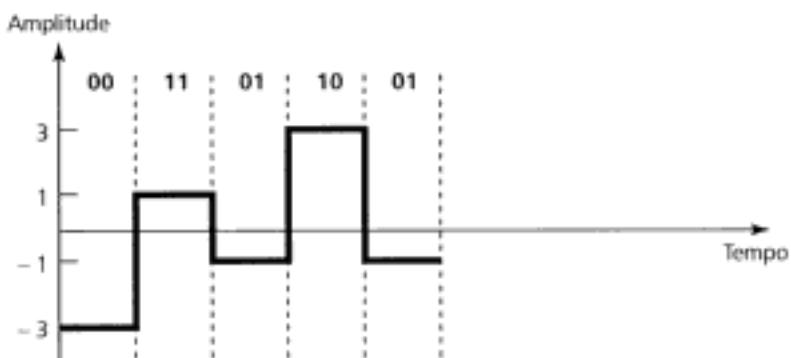


Figura 4.13 2B1Q.

MLT-3

O esquema **Multiline Transmission, Three Level** (MLT-3) é bastante semelhante ao NRZ-I, mas utiliza três níveis de sinal (+1, 0 e -1). Realiza as transições de um nível para o próximo no começo do *bit* 1. O sinal não realiza transições no começo de um *bit* 0. A Figura 4.14 mostra um exemplo de sinal MLT-3.

4.2 CODIFICAÇÃO DE BLOCOS

A **codificação de blocos** (*block coding*) foi desenvolvida para melhorar a *performance* da codificação de linha. O fato básico a respeito da codificação de linha é: necessitamos de algum tipo de redundância que assegure a sincronização. Além do que, precisamos ainda incluir *bits* adicionais (como veremos no Capítulo 10) para detectarmos erros de transmissão. A codificação de blocos pode satisfazer, e até mesmo estender, estes dois objetivos. A Figura 4.15 ilustra o procedimento de codificação de blocos.

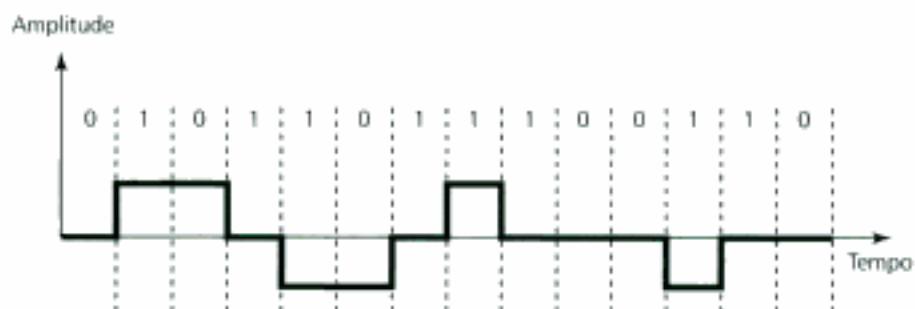


Figura 4.14 Sinal MLT-3.

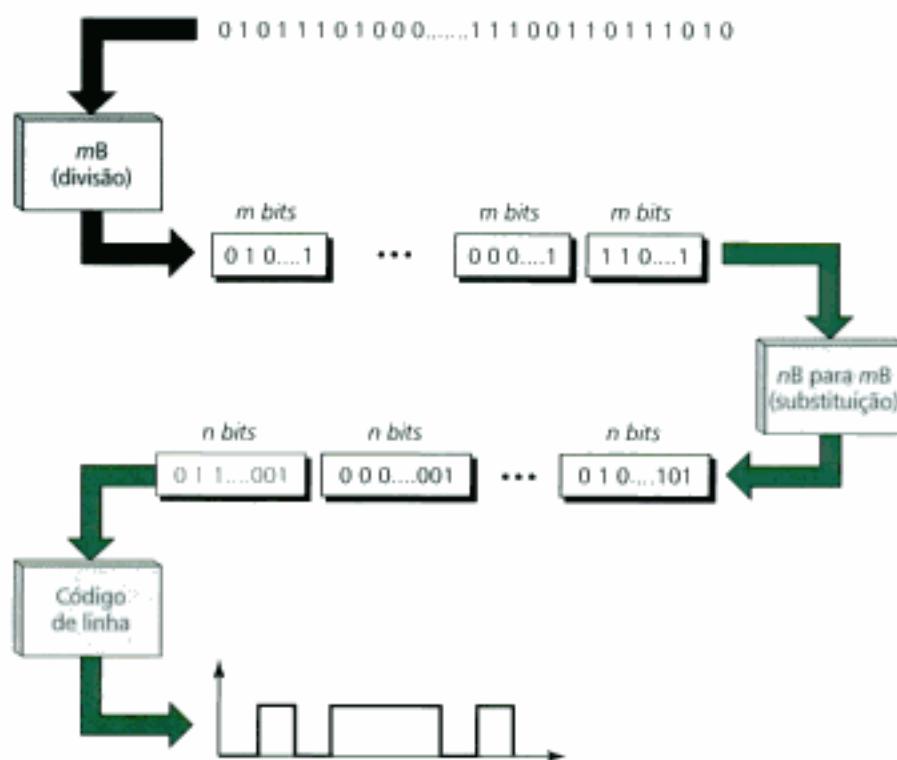


Figura 4.15 Codificação de blocos.

Etapas da Seqüência de Transformação

Neste método existem basicamente três passos a serem seguidos: divisão, substituição e codificação de linha.

1a Etapa: Divisão

Nesta etapa, a seqüência ou cadeia de *bits* é dividida em grupos de *m bits* de tamanho. Por exemplo, na codificação 4B/5B, a seqüência de *bits* original é dividida em grupos de 4 *bits*.

2a Etapa: Substituição

O coração da técnica de codificação de blocos é a etapa de substituição. Nela, realizamos a substituição de um código de *m-bits* por um grupo de *n-bits*. Por exemplo, na codificação 4B/5B substituímos um código de 5-*bits* por um grupo de 4-*bits*. Utilizando um bloco de 4-*bits* podemos formar 16 (2^4) grupos diferentes. De modo equivalente, através de um código de 5-*bits* podemos formar 32 (2^5) elementos codificáveis. Isto significa que alguns dos elementos do código de 5-*bits* podem ser

mapeados dentro do grupo de 4-bits. É claro que muitos dos elementos do código de 5-bits não terão nenhuma correspondência no grupo de 4-bits. Entretanto, podemos aplicar uma estratégia ou política para escolher os elementos do código de 5-bits que assegurem os mecanismos que facilitem a sincronização e a detecção de erros no receptor. A Figura 4.16 mostra como mapear metade dos elementos do código de 5-bits num grupo de 4-bits.

Para obter sincronização, podemos utilizar o código de 5-bits de um modo tal que, por exemplo, não tenhamos mais que três 0s ou 1s consecutivos na seqüência.

A codificação de blocos pode definitivamente resolver o problema de detecção de erros. O receptor pode facilmente detectar um erro de transmissão, visto que somente um subconjunto do código de 5-bits é utilizado. Se um ou mais bits no bloco for modificado de tal forma que um dos códigos não utilizados no subconjunto for recebido no receptor fica caracterizado erro de transmissão.

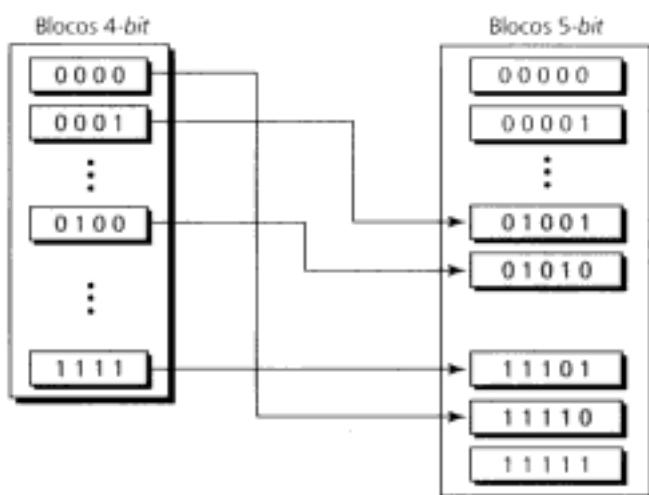


Figura 4.16 Substituição do bloco codificado.

3a Etapa: Codificação de linha

Após a etapa da substituição, podemos utilizar qualquer um dos esquemas de codificação de linha para criar um sinal codificado. Normalmente é escolhido um esquema de codificação de linha muito simples porque o procedimento de codificação em bloco proporciona bastante complexidade ao esquema de codificação de linha. Algumas vezes, como veremos, combinamos o segundo e terceiro passos (substituição e codificação de linha, respectivamente).

Alguns Blocos de Códigos

Discutiremos nesta seção alguns dos blocos de códigos mais comuns utilizados nos esquemas de codificação de blocos.

4B/5B

Como descrito acima, no código 4B/5B cada conjunto de 4-bits de dados é mapeado num código de 5-bits. A seleção do código de 5-bits é feita de forma tal que cada possibilidade contenha não mais que um 0 isolado e não mais que dois 0s agrupados. Logo, quando esses códigos de 5-bits são enviados em seqüência, não mais que três 0s consecutivos são encontrados no bloco. Os códigos de 5-bits são normalmente codificados em linha através do código NRZ-I. A Tabela 4.1 mostra o mapeamento 4B/5B. As seqüências codificadas para os caracteres de controle (terceira coluna) não seguem as regras de codificação 4B/5B.

8B/10B

Este esquema de mapeamento é similar ao esquema de codificação 4B/5B. A diferença é o tamanho da correspondência: um grupo de 8-bits de dados é substituído por outro de código de 10-bits. Is-

TABELA 4.1 Codificação 4B/5B

Dados	Código	Dados	Código
0000	11110	Q (Quiet)	00000
0001	01001	I (Idle)	11111
0010	10100	H (Halt)	00100
0011	10101	J (start delimiter)	11000
0100	01010	K (start delimiter)	10001
0101	01011	T (end delimiter)	01101
0110	01110	S (Set)	11001
0111	01111	R (Reset)	00111
1000	10010		
1001	10011		
1010	10110		
1011	10111		
1100	11010		
1101	11011		
1110	11100		
1111	11101		

so melhora a capacidade de detecção de erros no receptor. A tabela mostrando a correspondência de **codificação 8B/10B** é muito longa e não será mostrada aqui.

8B/6T

Vimos que tanto a codificação 4B/5B como a 8B/10B proporcionam ótima sincronização e capacidade de detecção de erro. O que não foi mencionado é que elas têm um preço: requerem uma largura de banda maior. Muitas vezes não dispomos dessa banda extra para podermos utilizá-las. A **codificação 8B/6T** foi criada para substituir um grupo de 8-bits por um código com seis símbolos. Isto significa que cada bloco de 8-bits dados é codificado em função de símbolos ternários (três níveis, +1, 0 e -1V). Um código de 8-bits pode representar até 256 possibilidades (2^8); um sinal de ternário com seis símbolos pode representar até 729 possibilidades (3^6). Outra vez o mapeamento não é completo. A codificação é escolhida de modo a facilitar o sincronismo e a capacidade de detecção de erro. O Apêndice D mostra a tabela completa de codificação 8B/6T. A Figura 4.17 ilustra um exemplo de codificação 8B/6T.

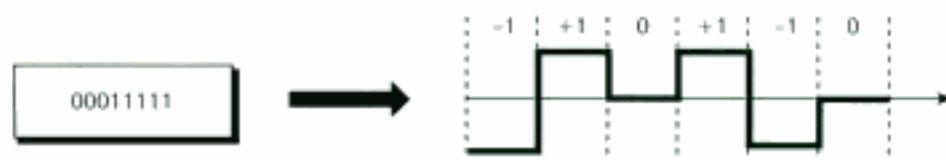


Figura 4.17 Exemplo de codificação 8B/6T.

4.3 AMOSTRAGEM

As codificações de linha e em bloco são utilizadas para converter dados binários em sinais digitais. Muitas vezes, entretanto, dispomos de dados na forma analógica, tal como um sinal de áudio. Voz e música, por exemplo, são de natureza analógica, de modo que, quando gravamos voz ou vídeo, criamos um sinal elétrico analógico. Se desejarmos armazená-los no computador ou transmiti-los digitalmente, devemos empregar um esquema de conversão denominado **amostragem**. Após a

amostragem do sinal analógico, podemos armazená-lo no computador ou utilizar um esquema de codificação de linha (às vezes uma combinação de codificação de linha e em blocos) para, em seguida, converter o sinal para digital e então transmiti-lo digitalmente.

A idéia da digitalização dos sinais analógicos nasceu nas companhias telefônicas. Para assegurar os serviços de longa distância, as companhias telefônicas devem ser capazes de transmitir sinais analógicos, transmitidos no canal de voz, através de extensos meios metálicos (os cabos). A intensidade dos sinais elétricos diminui à medida que eles viajam ao longo de condutores metálicos e a única saída é utilizar amplificadores ao longo da linha de transmissão para restaurar a intensidade do sinal. Entretanto, os amplificadores criam distorções nos sinais devido ao espectro de frequência, às mudanças de fase e aos ruídos intrínsecos do sistema. Como resultado, o sinal recebido não é uma réplica exata do sinal transmitido. Se você pudesse utilizar hoje o sistema telefônico de longa distância de algumas décadas atrás perceberia facilmente esse fenômeno.

A solução encontrada pelas companhias telefônicas foi digitalizar o sinal analógico no transmissor. Desse modo, o sinal é transmitido com um sinal digital e convertido de volta à forma analógica no receptor.

Como discutido no Capítulo 3, sinais digitais estão menos propensos a ruídos e distorções que os sinais analógicos. Uma pequena mudança num sinal analógico pode mudar substancialmente o sinal de voz recebido, mas quando esse sinal está na forma digital, uma simples troca de um 0 para 1 ou de um 1 para 0 pode provocar uma mudança menor (depende do caso).

Pulse Amplitude Modulation (PAM)

Um método de conversão de analógico para digital (A/D) muito difundido é o **Pulse Amplitude Modulation (PAM)**. Esta técnica de conversão A/D toma um sinal analógico, amostra-o e gera uma série de pulsos baseados no resultado da amostragem. O termo **amostragem** significa realizar a medição de valores instantâneos do sinal em intervalos iguais.

O método de amostragem utilizado na PAM é mais útil em outras áreas da engenharia do que nas comunicações de dados. Contudo, a modulação PAM é o fundamento de um método importante de conversão A/D, denominado **Pulse Code Modulation (PCM)**.

Na modulação PAM, o sinal original é amostrado em intervalos iguais (veja Figura 4.18). Essa modulação baseia-se numa técnica denominada *sample and hold* (amostra e mantém). Num dado momento, o valor instantâneo do sinal é lido e mantido (retido) por um breve instante de tempo. Desse modo, o valor amostrado é tomado apenas localmente no sinal analógico, de modo a amostrar apenas o valor no referido ponto da forma de onda.

Dissemos acima que a modulação PAM isolada não é muito útil nas comunicações de dados. De fato, o sinal amostrado ainda é um sinal analógico, não digital, visto que o resultado da amostragem é uma seqüência de pulsos de amplitudes variáveis. Para torná-lo digital, devemos modificá-lo através da técnica PCM.

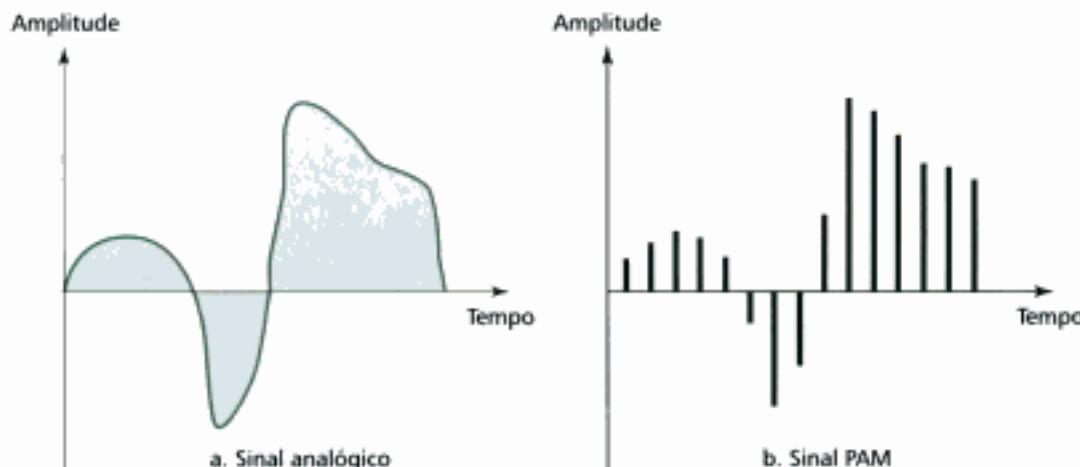


Figura 4.18 PAM.

A modulação PAM isolada não possui muitas aplicações, mas não é tão útil na comunicação de dados. Ela é o primeiro passo de outro método de conversão muito conhecido: a modulação PCM.

Pulse Code Modulation (PCM)

A modulação PCM modifica os pulsos criados pela PAM de modo a gerar um sinal totalmente digital. Para tanto, a PCM inicialmente quantiza os pulsos PAM. A **quantização** é um método que atribui valores inteiros, distribuídos numa determinada faixa, às amostras geradas na modulação PAM. O resultado da quantização aparece ilustrado na Figura 4.19.

A Figura 4.20 mostra um método simples de atribuição de intensidade e sinais às amostras quantizadas. Cada valor é traduzido em um código binário equivalente de 7-bits. O oitavo bit indica o sinal da conversão.

Em seguida, essa representação binária é transformada num sinal digital através de algum esquema de codificação de linha. A Figura 4.21 apresenta o resultado da modulação PCM de um sinal originalmente codificado em um sinal unipolar. Apenas os três primeiros valores amostrados são apresentados na figura.

A modulação PCM pode ser dividida em quatro processos separados: PAM, quantização, codificação binária e codificação de linha. A Figura 4.22 ilustra todo o processo na forma gráfica. A modulação PCM é o método de amostragem utilizado para digitalizar sinais de voz nas linhas de transmissão da maioria dos sistemas de telecomunicações (veja Capítulo 6).

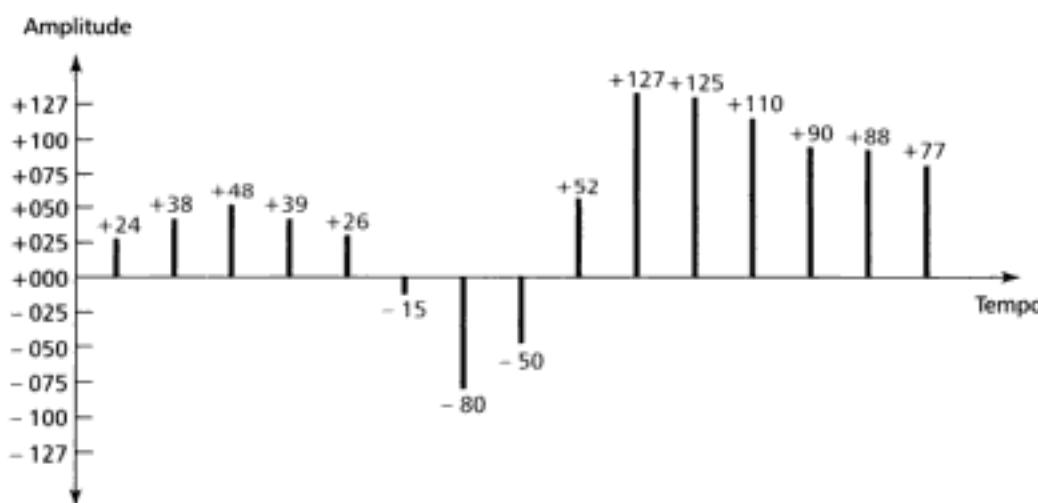


Figura 4.19 Quantização do sinal PAM.

+024	00011000	-015	10001111	+125	01111101
+038	00100110	-080	11010000	+110	01101110
+048	00110000	-050	10110010	+090	01011010
+039	00100111	+052	01101110	+088	01011000
+026	00011010	+127	01111111	+077	01001101

Sinal do bit
+ é 0 - é 1

Figura 4.20 Quantização usando sinal e magnitude.



Figura 4.21 PCM.

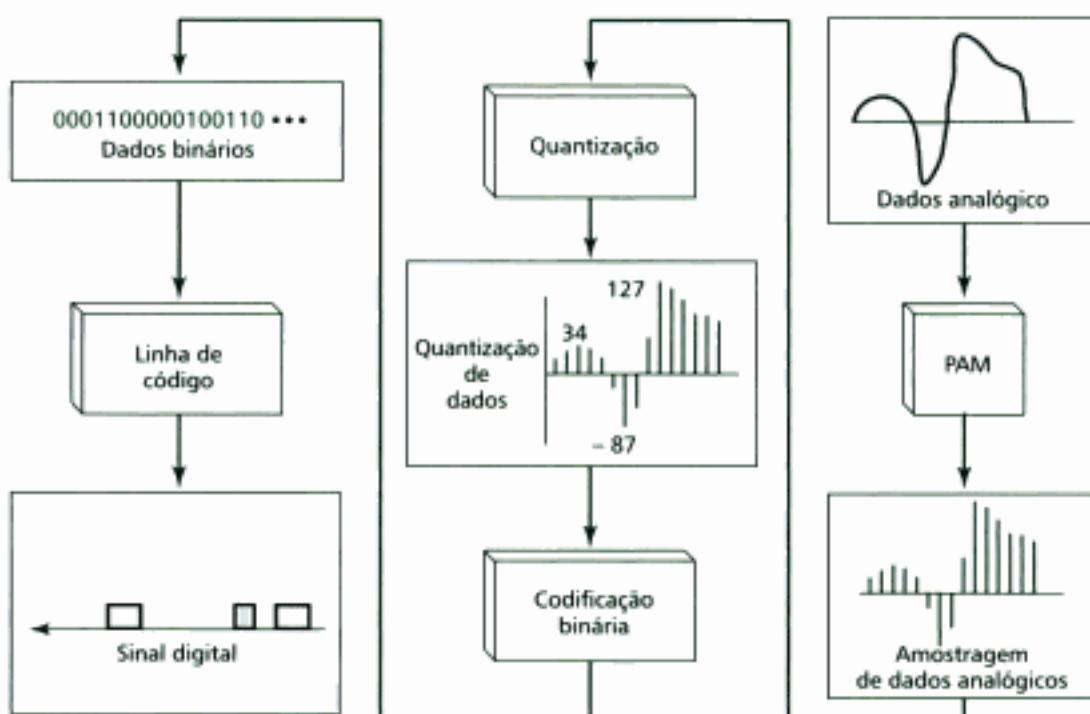


Figura 4.22 De sinal analógico para código digital PCM.

Taxa de Amostragem: Teorema de Nyquist

A precisão de qualquer reprodução digital de um sinal analógico depende do número de amostras realizadas no sinal original. Utilizando PAM e PCM, um sinal analógico original pode ser reproduzido exatamente se tomarmos uma quantidade infinita de amostras dele, ou então, pode ser reproduzido muito pobremente com apenas três amostras, uma para cada sinal exibido na forma de onda (+, - e 0). Obviamente, preferimos determinar um número entre esses dois extremos. Daí, vem a questão: quantas amostras do sinal analógico original são necessárias para que o dispositivo receptor possa reconstruir-lo?

Realmente, é notável o fato de serem necessárias relativamente poucas amostras para que o receptor consiga reconstruir um sinal analógico original. De acordo com o **teorema de Nyquist**, para assegurar a precisão da reprodução de um sinal analógico original, usando PAM, a **taxa de amostragem** deve ser pelo menos duas vezes a maior freqüência do sinal original. Assim, se quisermos realizar amostragens no canal de voz, cuja freqüência máxima é da ordem de 4000Hz, necessitaremos de uma taxa de amostragem de 8000 amostras por segundo.

De acordo com o teorema de Nyquist, a taxa de amostragem deve ser no mínimo duas vezes a mais alta freqüência do sinal original.

Uma taxa de amostragem de duas vezes uma freqüência de x Hz significa que o sinal deve ser amostrado a cada $1/2x$ segundos. Utilizando o exemplo do canal de voz anterior, significa que o sinal deve ser amostrado a cada 1/8000s. A Figura 4.23 ilustra o conceito.

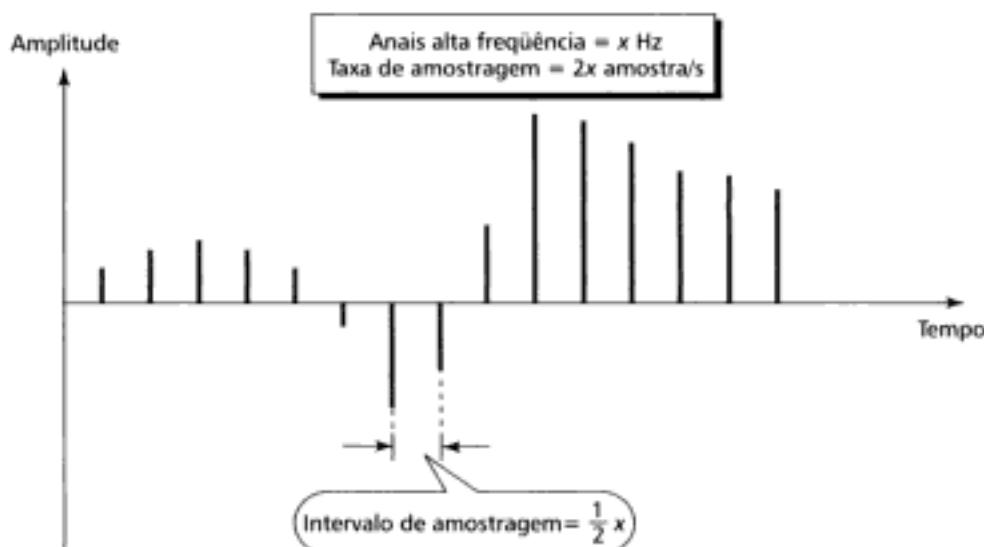


Figura 4.23 Teorema de Nyquist.

Note que é sempre possível deslocar a banda de freqüência de um sinal, centrada numa certa freqüência característica, antes de amostrá-lo e sem que o deslocamento modifique da banda do sinal. Nesse caso, a taxa de amostragem é duas vezes a largura de banda.

Exemplo 4

Qual é a taxa de amostragem de um sinal cuja largura de banda vale 1kHz (1kHz a 11kHz)?

Solução

A taxa de amostragem deve ser, no mínimo, duas vezes a mais alta freqüência no sinal. Logo,

$$\text{Taxa de amostragem} = 2 \times (11.000) = 22.000 \text{ amostras/segundo}$$

Quantos Bits por Amostra?

Resolvendo o problema da taxa de amostragem, precisamos determinar a quantidade de *bits* a serem transmitidos em cada amostra. O ponto fundamental é que isto depende do nível de precisão requerido. A quantidade de *bits* é escolhida de modo tal que o sinal original possa ser reproduzido com a precisão desejada no valor de amplitude.

Exemplo 5

Um sinal é amostrado. Cada amostra requer no mínimo 12 níveis de precisão (+0 a +5 e -0 a -5). Quantos *bits* serão necessários enviar em cada amostra?

Solução

Serão necessários 4 *bits*: 1 *bit* para representar o sinal e 3 *bits* para representar o valor (intensidade). Sabemos que 3 *bits* podem representar $2^3 = 8$ níveis (000 a 111), o que é mais do que necessitamos. A representação em 2 *bits* para os valores é descartada porque $2^2 = 4$, isso não é suficiente. Por fim, a representação em 4 *bits* para os valores é grande demais $2^4 = 16$.

Número de Bits por Segundo

Após determinar o número de *bits* por amostragem, podemos calcular o número de *bits* por segundo através da seguinte fórmula:

$$\text{Número de bits por segundo} = \text{taxa de amostragem} \times \text{número de bits por amostra}$$

Exemplo 6

Desejamos digitalizar a voz humana. Qual é o número de *bits* por segundo, assumindo 8 *bits* por amostra?

Solução

A voz humana normalmente tem freqüências compreendidas entre 0 e 4.000Hz. Desse modo, a taxa de amostragem é:

$$\text{Taxa de amostragem} = 4000 \times 2 = 8000 \text{ amostras/segundo}$$

Daí,

$$\text{Número de bits por segundo} = \text{taxa de amostragem} \times \text{número de bits por amostra}$$

$$\text{Número de bits por segundo} = 8000 \times 8 = 64.000 \text{ bps} = 64 \text{ kbps}$$

4.4 MODOS DE TRANSMISSÃO

Um dos princípios básicos que norteia a conexão de dispositivos a outro para a transmissão de dados é o cabeamento. Já o princípio básico que norteia o cabeamento é o tipo e a quantidade de cabos ou condutores necessários a garantir o fluxo de dados. Essa escolha geralmente leva a seguinte questão: devemos enviar um *bit* por vez ou enviar um conjunto de *bits* num mesmo intervalo de tempo? A transmissão de dados binários através de um *link* pode ser feita de modo serial ou paralelo. No modo serial, 1 *bit* é enviado por vez, isto é, a cada ciclo de *clock*. No modo paralelo muitos *bits* são enviados a cada ciclo de *clock*. Existe apenas um modo de transmissão paralela, enquanto que podemos transmitir dados seriais de duas formas diferentes: síncrona ou assíncrona.



Figura 4.24 Transmissão de dados.

Transmissão Paralela

Dados binários (pacotes de 0s e 1s) podem ser agrupados em conjuntos n *bits* cada. Assim como nós seres humanos concebemos e usamos na língua falada palavras, em vez de letras, os computadores produzem e se utilizam de dados em grupos de *bits*. Agrupando dados ele pode enviar e receber n *bits* de dados ao mesmo tempo, ao invés de um único *bit* por vez. Este método é denominado **transmissão paralela**.

O mecanismo da transmissão paralela de dados é conceitualmente muito simples: utiliza n veículos (meios) para enviar n *bits* ao mesmo tempo. Assim, cada *bit* tem um meio próprio para viajar e todos os n *bits* do grupo são enviados, de um dispositivo a outro, de uma só vez, a partir de um único ciclo de *clock*. A Figura 4.25 mostra como a transmissão paralela trabalha com um conjunto de 8 *bits*. Tipicamente, 8 fios são alojados num cabo com um conector em cada ponta.

A vantagem da transmissão paralela é a velocidade. A velocidade numa transmissão paralela é da ordem de n vezes a velocidade da transmissão serial, isto sendo todos os demais requisitos iguais nas duas formas de transmissão. Entretanto, existe uma desvantagem decisiva: o custo. A transmissão paralela requer n linhas ou canais de comunicação (fios no exemplo) para transmitir conjuntos de dados. Este fator limita a faixa de aplicação da transmissão paralela a curtas distâncias.



Figura 4.25 Transmissão paralela.

Transmissão Serial

Na **transmissão serial**, os *bits* são enviados um após o outro, isto é, em série ou formando uma fila. Assim, é necessário apenas um canal de comunicação para transportá-los, em vez de *n* canais idênticos, entre dois dispositivos de comunicação de dados (veja Figura 4.26).

A maior vantagem da transmissão serial sobre a transmissão paralela é que os custos da transmissão diminuem, aproximadamente, de um fator *n*.

Visto que a transmissão de dados dentro dos dispositivos acontece paralelamente, é necessário um dispositivo interno adicional no transmissor para converter os dados paralelos em serial (conversão paralela-serial) e outro no receptor, para restaurar a forma original dos dados, isto é, uma conversão serial-paralela.

A transmissão serial ocorre de dois modos: assíncrono e síncrono.



Figura 4.26 Transmissão serial.

Transmissão Assíncrona

A denominação **transmissão assíncrona** se deve ao fato da sincronização do sinal transmitido entre dispositivos não ser muito importante. Em vez disso, a informação é trocada entre os dispositivos através de um processo de sinalização de início e fim. Tão logo o dispositivo que possui dados a transmitir (transmissor) inicie o processo de sinalização, o dispositivo receptor pode se preparar para recuperar a informação transmitida, sem se preocupar com o ritmo segundo o qual a transmissão está acontecendo. O padrão de transmissão baseia-se no envio de caracteres. Cada agrupamento de *bits*, geralmente de 7 ou 8 *bits*, é enviado ao longo do *link* como um caractere mais a sinalização. O sistema que transmite os dados monta cada grupo independentemente, considerando o *link* sempre pronto para transmitir, ou seja, não havendo necessidade de um sincronismo entre as duas pontas do *link*.

Sem o processo de sincronização, o receptor não tem como ajustar o relógio para prever quando o próximo grupo chegará. O receptor fica, assim, sujeito a um sinal de alerta de início e fim de transmissão, que é feito no início e no fim de cada caractere. Este *bit*, geralmente um 0, é denominado ***start bit***. Para indicar ao receptor que o caractere chegou ao fim, um ou mais *bits* em nível 1 são adicionados no final do caractere. Esses *bits*, usualmente 1s, são denominados ***stop bits***. Desse modo, para que cada caractere seja transmitido são necessários enviar ao receptor de 9 a 11 *bits* ao todo, 7 ou 8 *bits* de informação (caractere) e 2 ou 3 *bits* de sinalização (1 *start bit* mais 1 ou 2 *stop bits*). Além disso, a informação transmitida pode ser seguida de uma condição de repouso ou marca de duração variável. Esse período de repouso pode representar o canal inativo (desocupado) ou pode ser formado por uma cadeia adicional de *stop bits* (sequência de *bits* em 1).

No modo de transmissão assíncrono, o processo inicia-se com 1 *start bit* em nível 0. Em seguida, vem um caractere, geralmente em 7 ou 8 *bits*, e finaliza com um 1 ou mais *stop bits* em nível 1. Pode ocorrer um período de repouso (marca) entre os caracteres transmitidos no link.

Os *bits* de *start/stop* e a condição de repouso alertam o receptor para o início e fim de cada caractere. Isso possibilita estabelecer sincronismo entre o receptor e o transmissor durante a comunicação de cada um dos caracteres. Este mecanismo é denominado assíncrono porque, antecedendo a transmissão de cada caractere, transmissor e receptor não precisam estar sincronizados. O sincronismo no receptor deve acontecer exatamente dentro de cada caractere recebido. Isto é, algum nível de sincronização é requerido, mas somente durante o intervalo de um caractere. Em suma, o receptor deve ser capaz de sincronizar o relógio a cada novo caractere recebido. Quando o receptor detecta um *start bit*, dispara um relógio e começa a contar os *bits* que chegam até ele. Recebidos *n bits*, o receptor procura um ou mais *stop bits*. Tão logo detecte-o(s), entra no modo de espera pelo próximo *start bit*.

Assíncrono significa “assíncrono no nível do caractere”, mas no nível de cada *bit* há sincronismo. O intervalo de sinalização de cada *bit* é o mesmo.

A Figura 4.27 é uma ilustração esquemática do modo de transmissão assíncrono. Neste exemplo, os *start bits* estão em nível 0, os *stop bits* estão em nível 1 e as marcas estão representadas pela condição de linha inativa em vez de *stop bits* adicionais.

Os *bits* adicionais (*start*, *stop* e marca) ao caractere tornam a comunicação assíncrona mais lenta que outras formas de transmissão passíveis de operação sem a adição de controle de informação*. Mas ela é barata e efetiva, duas vantagens que a tornam uma escolha particularmente atrativa em situações tais como na comunicação em baixas velocidades. Por exemplo, a conexão de um teclado a um computador é uma aplicação natural para o modo de transmissão assíncrono. Um usuário digita apenas um caractere por vez e o faz de maneira extremamente lenta, em termos de processamento de dados, deixando um intervalo de repouso imprevisível na linha, entre os caracteres digitados.

Transmissão Síncrona

Na transmissão síncrona, blocos de *bits* são combinados em longos quadros ou *frames*, e podem ser constituídos de muitos *bytes*. Porém, os *bytes* são introduzidos no link de transmissão sem a existência de nenhum intervalo de repouso entre eles. Cabe ao receptor a tarefa de separar os quadros em *bytes* e decodificar os propósitos deles. Noutras palavras, os dados são transmitidos nu-

* N. de R. T.: Isto geralmente tem relação com o alto índice percentual que mede a taxa de sinalização do canal (*overhead*) e a utilização ruim que a transmissão assíncrona faz do canal. Para se ter uma noção exata, cada caractere do código ASCII padrão tem 7 *bits*. Para transmiti-lo, necessitamos de 1 *start bit*, mais 1 ou 2 *stop bits* e, possivelmente, 1 *bit* de paridade no final do caractere. No pior caso, serão enviados 11 *bits* no todo, sendo 4 *bits* de sinalização. Nesse caso, o *overhead* da transmissão é $4/11 = 36,4\%$. Esse resultado pode parecer pouco importante. Pense no seguinte: se você tivesse que transmitir assincronicamente 1 *Mbyte* de dados em ASCII, qual seria o tamanho efetivamente transmitido no final?

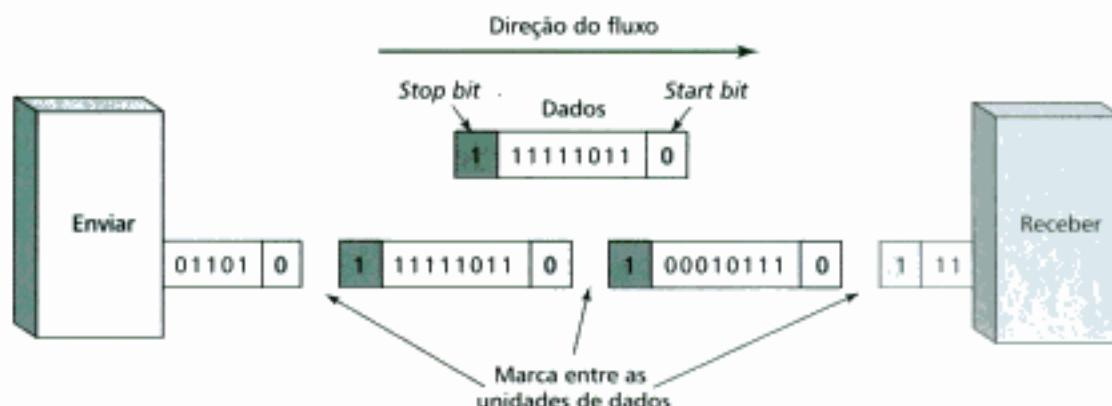


Figura 4.27 Transmissão assíncrona.

ma cadeia extensa e ininterrupta de 1s e 0s, e o receptor quebra a cadeia em *bytes* ou caracteres de modo a reconstruir a informação.

No modo de transmissão síncrono, enviamos *bytes* um após o outro sem *start/stop bits* ou intervalos de repouso. O receptor é responsável pelo agrupamento e pela interpretação dos *bytes* recebidos.

A Figura 4.28 mostra uma ilustração esquemática da transmissão síncrona. Os *bytes* foram desenhados com uma linha divisória entre eles. Na realidade, essas divisões não existem. O transmissor despeja os dados na linha num longo bloco de dados. Se o transmissor desejar enviar dados formando quadros separados, os intervalos entre quadros devem ser preenchidos com seqüências especiais de 0s e 1s para representar um intervalo inativo. O receptor conta os *bits* que ele recebe e agrupa-os em *bytes*.

Sem *start/stop bits* e intervalos de repouso da linha, parece não existir mecanismos que ajudem o dispositivo receptor a ajustar o sincronismo com os *bits* no meio do bloco de dados. Nesse caso, é muito importante o receptor manter um relógio de sincronismo confiável, porque a precisão da informação recebida depende da habilidade do dispositivo em manter a contagem dos *bits* que chegam ao receptor em ordem.

A vantagem óbvia da transmissão síncrona é a velocidade. Sem *bits* extras ou intervalos de repouso da linha a serem introduzidos no transmissor e removidos no receptor e, por extensão, com menos *bits* a serem transportados através do *link*, a transmissão síncrona é mais rápida se comparada à transmissão assíncrona. Por esta razão, tornou-se útil nas aplicações em altas velocidades, tais como a transmissão de dados entre computadores. A sincronização dos *bytes* é efetivada na camada de enlace (veja Capítulo 11).

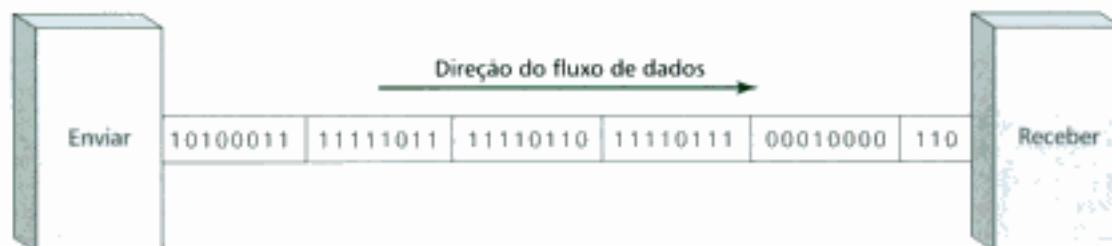


Figura 4.28 Transmissão assíncrona.

4.5 TERMOS-CHAVE

Alternate Mark Inversion (AMI)	Nível de sinal
Amostragem	NonReturn to Zero (NRZ)
Auto-sincronização	NonReturn to Zero, Invert (NRZ-I)
Bipolar n -Zero Substitution (BnZS)	NonReturn to Zero, Level (NRZ-L)
Codificação 2B1Q	Número de bits por segundo
Codificação 4B/5B	Pulse Amplitude Modulation (PAM)
Codificação 8B/10B	Pulse Code Modulation (PCM)
Codificação 8B/6T	Quantização
Codificação bipolar	Relógio de sincronismo
Codificação de blocos (<i>block coding</i>)	Return to Zero (RZ)
Codificação de linha	Start bit
Codificação Manchester	Stop bit
Codificação Manchester Diferencial	Taxa de amostragem
Codificação Multiline Transmission, <i>Three Level</i> (MLT-3)	Teorema de Nyquist
Codificação polar	Transmissão assíncrona
Codificação unipolar	Transmissão paralela
Componente DC	Transmissão serial
Nível de dados	Transmissão síncrona

4.6 RESUMO

- Codificação de linha é o processo de conversão de dados binários em sinal digital.
- A quantidade de valores diferentes permitidos num sinal é o nível de sinal. Ao número de símbolos utilizados para representar os dados damos o nome de níveis de codificação de dados.
- O número de bits por segundo é uma função do relógio de *clock* e do nível de codificação de dados.
- Os métodos de codificação de linha devem eliminar a componente DC e garantir um nível de sincronização entre o transmissor e o receptor.
- Os métodos de codificação de linha podem ser classificados como unipolar, polar e bipolar.
- As codificações NRZ, RZ, Manchester (simples e diferencial) são as técnicas mais comuns de codificação polar.
- AMI é um método de codificação bipolar muito difundido.
- A codificação de blocos é capaz de melhorar a *performance* da codificação de linha através de mecanismos de redundância e correção de erros.
- A codificação de blocos envolve o agrupamento dos bits, substituição e codificação de linha.
- 4B/5B, 8B/10B e 8B/6T são métodos muitos comuns de codificação de blocos.
- Conversão de analógico para digital (A/D) é utilizada na modulação PCM (Pulse Code Modulation).
- PCM envolve amostragem, quantização e codificação de linha.
- O teorema de Nyquist prova que a taxa de amostragem de um sinal deve ser, no mínimo, duas vezes a freqüência da componente de freqüência mais elevada no sinal original.
- Transmissão digital pode acontecer nos modos serial ou paralelo.
- Numa transmissão paralela, um grupo de bits é enviado simultaneamente, sendo cada bit enviado numa linha em separado (canal individual).
- Numa transmissão serial, há apenas um canal e os bits devem ser enviados seqüencialmente.
- A transmissão serial pode acontecer nos modos síncrono ou assíncrono.
- Numa transmissão serial assíncrona, cada caractere (grupo de 7 ou 8 bits) é sinalizado através de *start* e *stop bits*. Pode haver um intervalo de repouso de comprimento variável entre os caracteres.
- No modo de transmissão síncrono, os bits são enviados formando uma cadeia contínua, sem *start/stop bits* e intervalos de repouso. A recuperação da informação contida nas cadeias de bytes é de responsabilidade do receptor.

4.7 PRATIQUE OS CONHECIMENTOS ADQUIRIDOS

Questões de Revisão

1. Cite o nível de sinal para um dos métodos de codificação de linha discutidos (NRZ, RZ, etc.).
2. O que é a componente DC?
3. O número de *bits* por segundo pode ou não ser menor que a freqüência do relógio de *clock*? Explique.
4. Por que o sincronismo é um problema na comunicação de dados?
5. Qual(is) é(são) a(s) diferença(s) entre NRZ-L e NRZ-I?
6. Qual é a maior desvantagem da codificação NRZ? De que maneira a codificação RZ tenta resolver o problema?
7. Compare as codificações RZ e AMI.
8. Quais são os três passos da codificação de blocos?
9. De que forma a codificação de blocos pode ser útil na sincronização?
10. De que forma a codificação de blocos pode ser útil na detecção de erro?
11. Discorra sobre o relacionamento entre a taxa de amostragem e o sinal recebido.
12. Discorra sobre o relacionamento entre o número de *bits* distribuídos em cada amostra e o sinal recebido.
13. O que diz o teorema de Nyquist?
14. Explique os dois modos de transmissão de dados binários através de um *link* (canal).
15. Quais são as vantagens e desvantagens da transmissão paralela de dados?
16. Compare os dois métodos de transmissão serial. Comente as vantagens e desvantagens de cada um.

Questões de Múltipla Escolha

17. As codificações unipolar, bipolar e polar são tipos de codificação de _____.
 - Linha
 - Bloco
 - NRZ
 - Manchester
18. Se um símbolo é composto de 3 *bits*, há _____ níveis de codificação de dados.
 - 2
 - 4
 - 8
 - 16
19. O relógio de sincronismo é sempre _____ número de *bits* por segundo.
 - Maior que o
 - Menor que o
 - Maior do que ou igual ao
 - Menor do que ou igual ao
20. A codificação _____ possui uma transição no meio de cada *bit*.
 - RZ
 - Manchester
 - Manchester Diferencial
 - Todas acima
21. A codificação _____ exibe uma transição no início de cada *bit* 0.
 - RZ
22. PCM é um exemplo de conversão _____.
 - Digital para digital
 - Digital para analógico
 - Analógico para analógico
 - Analógico para digital
23. Se o espectro de freqüência de um sinal tem largura de banda 500Hz, com a maior freqüência na banda em 600Hz, de acordo com o teorema de Nyquist qual deve ser a taxa de amostragem desse sinal?
 - 200 amostras/segundo
 - 500 amostras/segundo
 - 1000 amostras/segundo
 - 1200 amostras/segundo
24. O teorema de Nyquist especifica a taxa de amostragem mínima como sendo _____.
 - Igual à menor freqüência de um sinal
 - Igual à maior freqüência de um sinal
 - Duas vezes a largura de banda de um sinal
 - Duas vezes a mais alta freqüência de um sinal

25. Um dos fatores que asseguram a precisão da reconstrução de um sinal PCM é a _____.
- Largura de banda do sinal
 - Freqüência da portadora
 - Quantidade de *bits* usados na quantização
 - Baud Rate
26. Que tipo de codificação tem sempre um valor médio diferente de zero?
- Unipolar
 - Polar
 - Bipolar
 - Todas acima
27. Qual dos seguintes métodos de codificação não assegura a sincronização?
- NRZ-L
 - RZ
 - NRZ-I
 - Manchester
28. Que método de codificação utiliza alternâncias de 1s, positiva e negativamente?
- NRZ-I
 - RZ
 - Manchester
 - AMI
29. Na modulação PCM, ocorre uma conversão analógica para _____.
- Analógico
 - Digital
 - QAM
 - Diferencial
30. Se os valores máximo e mínimo de um PCM são +31 e -31, quantos *bits* devemos utilizar no código de representação binário?
- 4
 - 5
 - 6
 - 7
31. A codificação RZ requer _____ níveis de sinal.
- Dois
 - Três
 - Quatro
 - Cinco
32. Qual dos seguintes níveis de quantização resultam numa reprodução mais fidedigna de um sinal?
- 2
 - 8
 - 16
 - 32
33. Que técnica de codificação tenta resolver o problema da perda de sincronização devido às longas cadeias de 0s?
- Bnzs
 - NRZ
 - AMI
 - (a) e (b)
34. Codificação de blocos favorece a _____ no receptor.
- Sincronização
 - Detecção de erros
 - Atenuação
 - (a) e (b)
35. Na transmissão _____, os *bits* são transmitidos simultaneamente, cada qual num devido meio ou canal.
- Serial assíncrona
 - Serial síncrona
 - Paralela
 - (a) e (b)
36. Na transmissão _____, os *bits* são transmitidos através de um único canal, um de cada vez.
- Serial assíncrona
 - Serial síncrona
 - Paralela
 - (a) e (b)
37. Na transmissão _____, um *start bit* e um *stop bit* enquadram um caractere.
- Serial assíncrona
 - Serial síncrona
 - Paralela
 - (a) e (b)
38. Na transmissão assíncrona, o intervalo de repouso da linha é _____.
- Fixo
 - Variável
 - Uma função da taxa de transmissão de dados
 - Zero
39. Transmissão síncrona não possui _____.
- Um *start bit*
 - Um *stop bit*
 - Intervalos entre *bytes*
 - Todas acima

Exercícios

40. Se o número de *bits* por segundo é 1000bps, quantos *bits* podem ser enviados em 5s? E em 1/5 s? Quantos em 100ms?
41. Considere uma cadeia de dados feita a partir de dez 0s. Codifique essa cadeia usando os esquemas de codificação abaixo. Quantas transições verticais você pode encontrar em cada esquema?
- Unipolar
 - NRZ-L
 - NRZ-I
 - RZ
 - Manchester
 - Manchester Diferencial
 - AMI
42. Repita o Exercício 41 para uma cadeia de dez 1s.
43. Repita o Exercício 41 para uma cadeia alternada de dez 0s e 1s.
44. Repita o Exercício 41 para uma cadeia de três 0s, seguida de dois 1s, seguida de dois 0s e outros três 1s.
45. A Figura 4.29 é a codificação unipolar de uma cadeia de dados. Que cadeia gerou essa codificação?
46. A Figura 4.30 é a codificação NRZ-L de uma cadeia de dados. Que cadeia gerou essa codificação?
47. Repita o Exercício 46 se a Figura 4.30 representa a codificação NRZ-I de uma cadeia de dados.
48. A Figura 4.31 é a codificação RZ de uma certa cadeia de dados. Que cadeia gerou essa codificação?
49. A Figura 4.32 é a codificação Manchester de uma cadeia de dados. Que cadeia gerou essa codificação?
50. Repita o Exercício 49 se a Figura 4.32 representa a codificação Manchester Diferencial de uma cadeia de dados.
51. A Figura 4.33 é a codificação AMI de uma certa cadeia de dados. Que cadeia gerou essa codificação?
52. Quantos níveis de codificação de dados há em cada um dos seguintes métodos?
- Unipolar
 - NRZ-L
 - NRZ-I
 - RZ
 - Manchester
 - Manchester diferencial

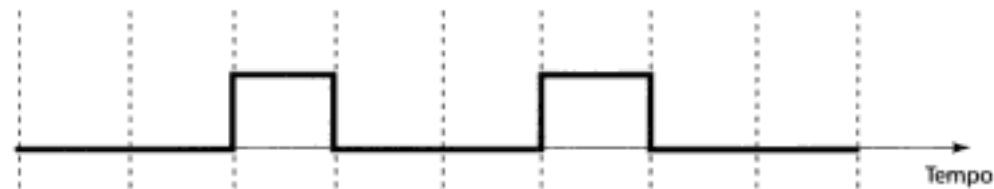


Figura 4.29 Exercício 45.

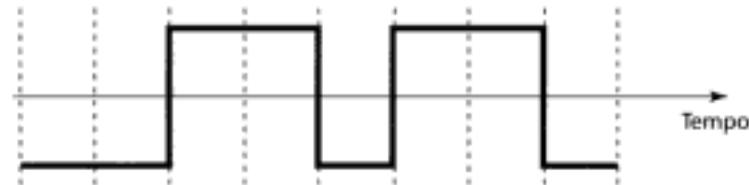


Figura 4.30 Exercícios 46 e 47.

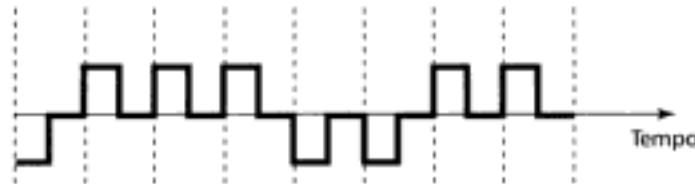


Figura 4.31 Exercício 48.



Figura 4.32 Exercícios 49 e 50.

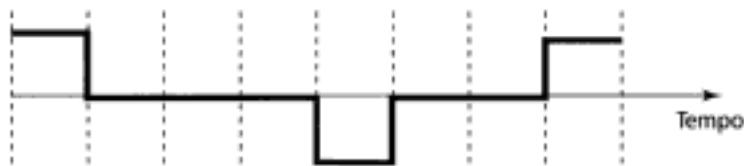


Figura 4.33 Exercícios 51 e 52.

53. Qual é a taxa de amostragem numa modulação PCM, se a faixa de freqüências varia de 1000 a 4000Hz?
54. Utilizando o teorema de Nyquist, determine a taxa de amostragem para os seguintes sinais analógicos.
 - a. Um sinal analógico com uma largura de banda de 2000Hz.
 - b. Um sinal analógico com freqüências variando entre 2000 e 6000Hz.
 - c. Um sinal que é uma linha horizontal num gráfico no domínio do tempo.
 - d. Um sinal que é uma linha vertical num gráfico no domínio da freqüência.
55. Se um sinal é amostrado 8000 vezes por segundo, qual é o intervalo entre cada amostra?
56. Se o intervalo entre duas amostras num sinal digitalizado é $125\mu\text{s}$, qual é a taxa de amostragem?
57. Um sinal é amostrado. Cada amostra é representada em quatro níveis. Quantos bits são necessários para representar cada amostra? Se a taxa de amostragem é 8000 amostras por segundo, qual é o número de bits por segundo?
58. Se desejarmos transmitir assincronamente 1000 caracteres ASCII (ver Apêndice A), qual é o número mínimo de bits extras necessários? Qual é a eficiência (*overhead*) da transmissão em porcentagem?

Transmissão Analógica

No Capítulo 3 discutimos as vantagens e desvantagens das transmissões analógica e digital. Vimos que a transmissão digital é bastante interessante, mas necessita de um canal passa-baixas com uma largura de banda relativamente alta. Também foi mencionado que a transmissão analógica é a única escolha possível se dispusermos apenas de um canal de natureza passa-banda. A transmissão digital foi abordada no Capítulo 4; neste capítulo trataremos as técnicas de transmissão analógica.

A técnica de converter sinais analógicos, ou dados binários, em outro sinal analógico de modo a transmiti-lo em um canal passa-banda é denominada *modulação*. Neste capítulo, trataremos primeiramente as técnicas de modulação de dados binários. Em seguida, discutiremos os modems, dispositivos que de fato possibilitam fazer algum tipo de modulação. Finalmente, mostraremos os esquemas de modulação de sinais analógicos em canais do tipo passa-baixas.

5.1 MODULAÇÃO DE DADOS DIGITAIS

A modulação de dados binários ou **modulação digital** é o processo que permite modificar uma ou mais características de um sinal analógico baseado na informação contida num sinal digital (0s e 1s). Por exemplo, quando você transmite dados de um computador para outro dispositivo através da rede de telefonia pública, os dados originais estão na forma digital dentro do computador, mas os meios metálicos das linhas telefônicas podem transportar apenas sinais analógicos. Desse modo, devemos converter (modular) esses dados para analógico antes de transmiti-los na linha telefônica. Os dados digitais devem ser modulados em um sinal analógico de acordo com regras que permitam distinguir claramente entre os valores 0s e 1s do sinal original. A Figura 5.1 mostra um bloco genérico capaz de implementar a modulação de um sinal binário de entrada para um sinal analógico na saída.

Dos muitos mecanismos de modulação de dados, discutiremos somente aqueles mais úteis na comunicação de dados.

Conforme foi mencionado no Capítulo 3, um sinal senoidal é definido através de três características: *amplitude*, *frequência* e *fase*. Quando variamos qualquer uma dessas características criamos uma versão diferente desse sinal. Se usarmos o sinal senoidal original para representar o binário 1, então, modificando o sinal podemos representar o binário 0 ou vice-versa. Assim, a modificação de algum dos aspectos de um sinal elétrico permite representar convenientemente os dados binários. Qualquer uma das três características listadas acima pode ser alterada. Isto possibilita pelo menos



Figura 5.1 Modulação digital-analógico.

três mecanismos para modulação de dados em sinais analógicos, que são: Amplitude Shift Keying (ASK), Frequency Shift Keying (FSK) e Phase Shift Keying (PSK). Além desses, há um quarto mecanismo que combina variações tanto na fase quanto na amplitude denominado Quadrature Amplitude Modulation (QAM)*. Como veremos, a modulação QAM é mais complexa que as outras três, mas apresenta resultados muito superiores, sendo por isso a preferida nos modems modernos (Figura 5.2).

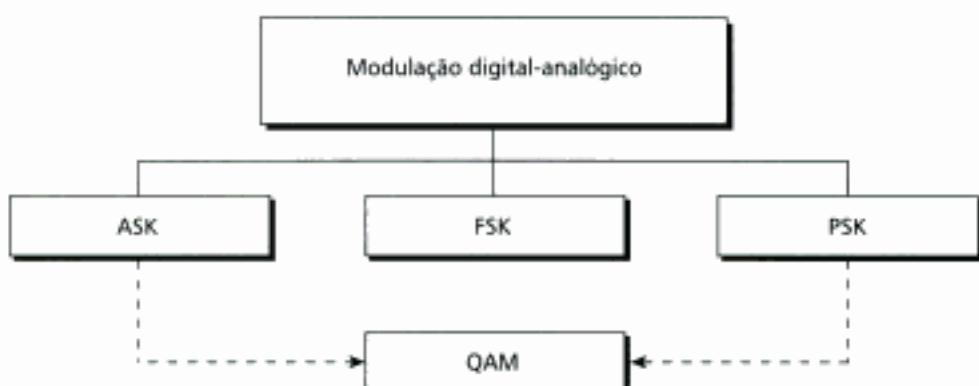


Figura 5.2 Tipos de modulação digital-analógico.

Aspectos da Conversão Digital para Analógico

Antes de iniciarmos a discussão específica dos métodos de modulação de dados, devemos definir os seguintes conceitos básicos: taxa de transmissão, taxa de modulação e portadora.

Taxa de Transmissão versus Taxa de Modulação

Dois termos freqüentemente encontrados na comunicação de dados são *taxa de transmissão (bit rate)* e *taxa de modulação (baud rate)*. A **taxa de transmissão** representa o número de *bits* transmitidos num intervalo de tempo igual a 1s. A taxa de transmissão é medida em *bits* por segundo (bps). A **taxa de modulação** refere-se à quantidade de modulações (sinalizações) realizadas durante 1s. A taxa de modulação é medida em **baud**. Numa única modulação podem estar representados 1 ou mais *bits*, dependendo da quantidade de **símbolos** enviados num único intervalo de modulação. Quando o que está em pauta é a eficiência da comunicação de dados, a taxa de transmissão de dados é mais importante porque ela informa o volume de informação binária transmitida. Entretanto, na transmissão de dados, volta e meia é interessante descobrir o quão eficientemente podemos mover dados de um lugar a outro, sem perdas de blocos de informação. Quanto menor o número de modulações na linha mais eficiente é o sistema de transmissão e menor é a largura de banda requerida para transmitir uma certa quantidade de *bits*. Nesse caso, a taxa de modulação em baud é um parâmetro mais interessante, pois ela determina a largura de banda real necessária à transmissão de um sinal.

* N. de R. T.: Optamos por manter em inglês os métodos de modulação. Entretanto, as melhores traduções são: ASK: Modulação por deslocamento de amplitude; FSK: Modulação por deslocamento de freqüência; PSK: Modulação por deslocamento de fase; QAM: Modulação por deslocamento de fase e amplitude

A taxa de transmissão é igual à taxa de modulação vezes o número de *bits* por símbolo representado em cada sinal. A taxa de transmissão é sempre maior do que ou igual à taxa de modulação.

A taxa de transmissão é o número de *bits* por segundo. A taxa de modulação é a quantidade de modulações por segundo. A taxa de transmissão é sempre maior do que ou igual à taxa de modulação.

Exemplificando os conceitos relativos as duas taxas. A taxa de modulação é o análogo a transportar um carro enquanto que a taxa de transmissão é o análogo a transportar pessoas. Um carro geralmente leva uma ou mais pessoas. Se 1000 carros vão de um ponto a outro através de uma rodovia transportando apenas uma pessoa (o motorista), então 1000 pessoas são transportadas. Entretanto, se cada carro transportar 4 pessoas (capacidade máxima), então são transportadas 4000 pessoas ao todo. Perceba que é o número de carros e não o de pessoas que determina o tráfego e, assim, a necessidade de uma rodovia com mais pistas. Similarmente, a taxa de modulação determina a largura de banda requerida, não o número de *bits* por segundo.

Exemplo 1

Um sinal analógico transporta 4 *bits* por símbolo. Se a linha é sinalizada (modulada) 1000 vezes por segundo, determine as taxas de transmissão e de modulação.

Solução

$$\text{Taxa de modulação} = \text{número de sinalizações por segundo} = 1000 \text{ baud/s}$$

$$\text{Taxa de transmissão} = \text{taxa de modulação} \times \text{quantidade de bits por símbolo} = 1000 \times 4 = 4000 \text{ bps}$$

Exemplo 2

A taxa de transmissão de um determinado sinal é 3000 bps. Se cada símbolo corresponde a 6 *bits*, qual é a taxa de modulação desse sinal?

Solução

$$\text{Taxa de modulação} = \text{Taxa de transmissão} + \text{número de bits por símbolo} = 3000 + 6 = 500 \text{ baud/s}$$

Portadora

Numa transmissão analógica, o dispositivo transmissor produz um sinal de alta freqüência que funciona como suporte para o sinal de informação. Este sinal suporte é denominado **portadora** ou **freqüência portadora**. O dispositivo receptor é sintonizado na freqüência da portadora que ele espera receber do transmissor. A informação digital modula então o sinal da portadora modificando uma ou mais características dela (amplitude, freqüência ou fase). Este tipo de modificação é denominada *shift keying* e o sinal da informação é chamado *sinal modulante*.

Amplitude Shift Keying (ASK)

Na técnica **ASK**, a intensidade ou amplitude do sinal da portadora varia de modo a representar a informação binária 0 ou 1. Tanto a freqüência quanto a fase permanecem constantes enquanto a amplitude sofre variações. Os níveis de tensão que representam os níveis 0 e 1 dependem do sistema em questão. O tempo de um *bit* é o intervalo de tempo que define o valor de um *bit*. Nesse intervalo, a amplitude de pico do sinal permanece constante e, como dissemos, o valor depende do *bit* representado (0 ou 1). A Figura 5.3 ilustra a idéia da modulação ASK.

Infelizmente, a modulação ASK é extremamente suscetível à interferência provocada por ruídos. Na maioria das vezes, o termo *ruido* refere-se às tensões indesejadas (induzidas ou geradas) dentro da linha de transmissão cuja origem se dá por diversos fenômenos, tal como o calor, a indução eletromagnética, dentre outras fontes. Estes níveis de tensão combinam-se com sinal da portadora, modificando a amplitude desse sinal. Um ruído pode corromper o sinal, modificando facilmente um sinal de nível 0 para outro em nível 1 e vice-versa. O fato da modulação ASK basear-se somente na amplitude do sinal para modular os níveis binários é um fator problemático nesse tipo de modulação. O problema principal é: ruídos geralmente afetam a amplitude do sinal que está sendo transmitido; assim, a modulação ASK é o método de modulação mais afetado por ruídos.

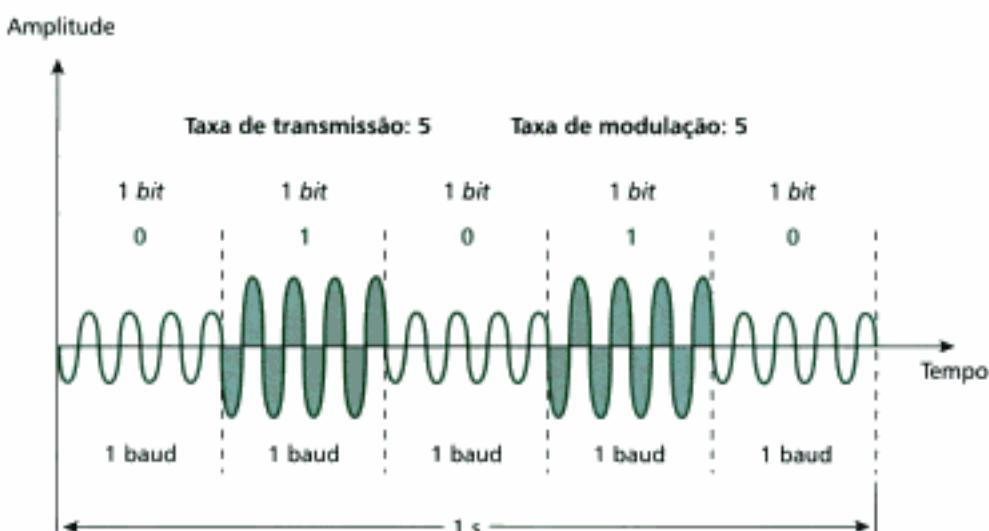


Figura 5.3 ASK.

Uma técnica de modulação ASK muito conhecida é denominada *On/Off Keying* (OOK). Na modulação OOK, o nível 0 ou nível 1 é representado por um nível de tensão de 0V. A maior vantagem da OOK é a redução na quantidade de energia requerida para transmitir informação.

Largura de Banda ASK

Como você pode verificar no Capítulo 3, a largura de banda de um sinal é a faixa total de freqüências ocupadas pelo sinal dentro do espectro de freqüências. Quando decomponemos em componentes de freqüência um sinal modulado em ASK, normalmente obtemos um espectro de freqüências bastante simples. Entretanto, apenas as freqüências situadas entre $f_c - N_{baud}/2$ e $f_c + N_{baud}/2$ são significativas, onde f_c é a freqüência central da portadora (veja Fig. 5.4).

A largura de banda requerida pela modulação ASK pode ser determinada através da fórmula:

$$BW = (1 + d) \times N_{baud}$$

onde BW é a largura de banda, em Hertz, N_{baud} é a taxa de modulação, em baud, e d é um fator relacionado ao processo de modulação (cujo valor mínimo é 0).

Como você pode constatar, a largura de banda mínima requerida para a transmissão de um sinal ASK deve igualar-se à taxa de modulação.

Embora nessa modulação a portadora seja única, o processo de modulação produz um sinal complexo que é a combinação de muitos sinais simples, cada qual com uma freqüência diferente.

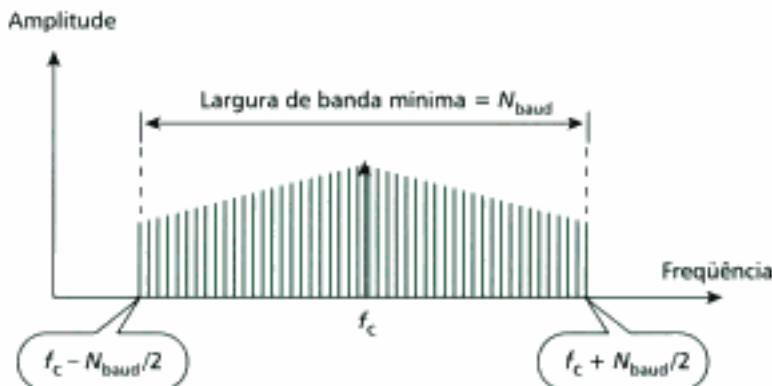


Figura 5.4 Relação entre largura de banda e taxa de modulação ASK.

Exemplo 3

Determine a largura de banda para um sinal ASK que está sendo transmitido a 2000 bps. O modo de transmissão na linha é *half-duplex*.

Solução

Na modulação ASK não há distinção entre a taxa de transmissão e de taxa de modulação. Desse modo, a taxa de modulação vale 2000 baud. Um sinal ASK requer uma largura de banda mínima igual à taxa de modulação. Assim, a largura de banda mínima vale 2000 Hz.

Exemplo 4

Para um dado sinal ASK de largura de banda de 5000 Hz, quais são as taxas de transmissão e de modulação?

Solução

A taxa de modulação ASK é idêntica à largura de banda, isto é, a menos de unidade da medida, 5000 baud/s. Como essas taxas são idênticas na modulação ASK, a taxa de transmissão também vale 5000 bps.

Exemplo 5

Dada uma largura de banda de 10 kHz (1 kHz a 11 kHz), desenhe o diagrama ASK/*full-duplex* do sistema. Determine as freqüências das portadoras e as larguras de banda em cada direção. Assuma também que não existe separação entre as bandas nas duas direções.

Solução

Para a modulação ASK/*full-duplex*, a largura de banda em cada direção vale:

$$BW = \frac{10 \text{ kHz}}{2} = 5 \text{ kHz}$$

As freqüências das portadoras podem ser escolhidas no meio de cada banda (veja Fig. 5.5).

$$f_{c(\text{direto})} = 1 \text{ kHz} + \frac{5 \text{ kHz}}{2} = 3,5 \text{ kHz}$$

$$f_{c(\text{inverso})} = 11 \text{ kHz} - \frac{5 \text{ kHz}}{2} = 8,5 \text{ kHz}$$

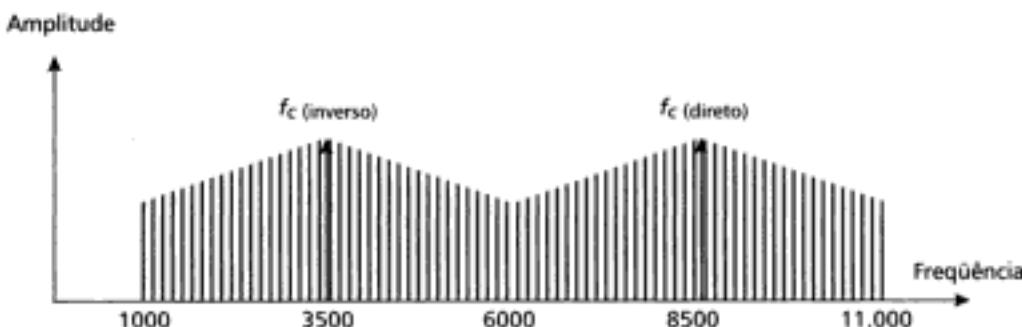


Figura 5.5 Solução para o Exemplo 5.

Frequency Shift Keying (FSK)

Na técnica **FSK**, a freqüência do sinal da portadora varia de modo a representar os níveis binários 0 ou 1. A freqüência do sinal é mantida constante durante cada intervalo de *bit*, mas o valor da freqüência em cada intervalo depende do *bit* representado; tanto o valor de amplitude quanto a fase permanecem inalterados em cada intervalo de *bit*. A Figura 5.6 ilustra conceitualmente a modulação FSK.

Na técnica FSK muitos dos problemas com ruídos são eliminados, visto que o dispositivo receptor é posicionado a “olhar” variações específicas na freqüência num certo intervalo de tempo. Desse modo, o receptor é capaz de ignorar surtos ou picos de tensão. Dentre os fatores que limitam o uso da modulação FSK aparece a capacidade física da portadora, em termos de banda.

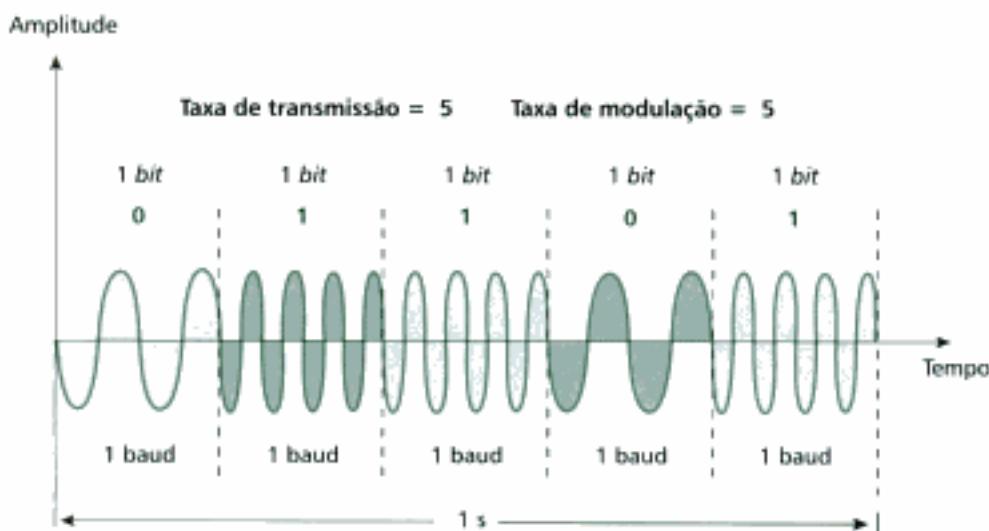


Figura 5.6 FSK.

Largura de Banda FSK

Embora a técnica FSK represente dados efetuando deslocamentos entre duas portadoras, é mais fácil analisá-la através de duas freqüências coexistindo num mesmo canal. Assumiremos que o espectro FSK é uma combinação de dois espectros ASK centrados nas freqüências f_{c0} e f_{c1} . Logo, a largura de banda exigida para a transmissão FSK é igual à taxa de modulação do sinal mais o deslocamento de freqüência, ou seja, a diferença entre as duas freqüências portadoras: $BW = N_{baud} + f_{c1} - f_{c0}$. Veja a Figura 5.7.

Embora na transmissão FSK existam apenas duas portadoras, o processo de modulação produz um sinal composto que é a combinação de muitos sinais, cada qual com uma freqüência diferente.

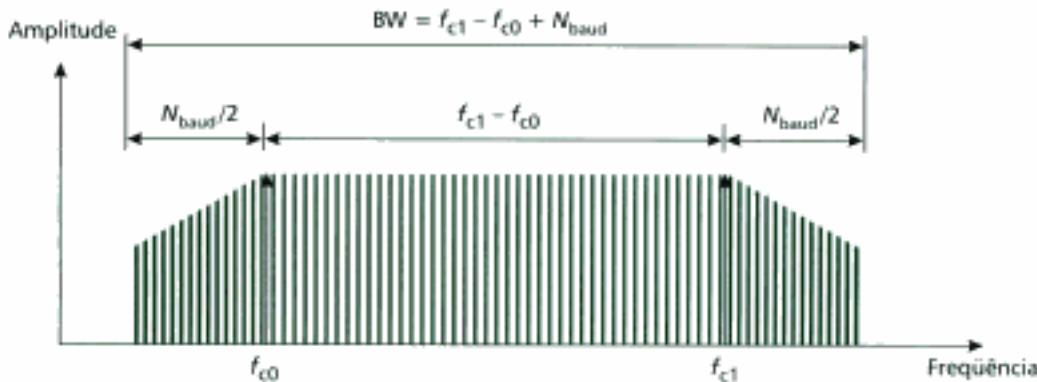


Figura 5.7 Relação entre taxa de modulação e largura de banda FSK.

Exemplo 6

Determine a largura de banda mínima para transmitir um sinal FSK a 2000 bps. Assuma que a transmissão ocorre no modo *half-duplex* e que as portadoras estão separadas de 3 kHz.

Solução

Para FSK, se f_{c1} e f_{c0} são as freqüências das portadoras, então:

$$BW = N_{baud} + f_{c1} - f_{c0}$$

Além disso, a taxa de modulação é igual à taxa de transmissão. Assim,

$$BW = N_{baud} + f_{c1} - f_{c0} = 2000 + 3000 = 5000 \text{ Hz} = 5 \text{ kHz}$$

Exemplo 7

Determine a taxa de transmissão máxima de um sinal FSK se a largura de banda do meio vale 12 kHz e a diferença entre as duas portadoras é 2 kHz. Assuma transmissão no modo *full-duplex*.

Solução

Como a transmissão ocorre no modo *full-duplex*, em cada direção é reservada uma banda de 6 kHz. Se f_{c1} e f_{c2} são as freqüências das portadoras, então:

$$\text{BW} = N_{\text{band}} + f_{c1} - f_{c2}$$

$$\text{Taxa de modulação} = \text{BW} - (f_{c1} - f_{c2}) = 6\text{kHz} - 2\text{kHz} = 4000 \text{ baud/s}$$

Outra vez a taxa de transmissão é numericamente igual à taxa de modulação, isto é, 4000 bps.

Phase Shift Keying (PSK)

Na técnica **PSK**, a fase da portadora é variada de modo a representar os níveis 0 ou 1. Além disso, tanto a amplitude quanto a freqüência permanecem constantes enquanto a fase estiver variando. Por exemplo, se partirmos do pressuposto que uma fase igual a 0° representa o binário 0, então podemos variar a fase para 180° de modo a enviar o binário 1. O importante é que a fase do sinal permaneça inalterada durante a representação do bit 0 ou 1. A Figura 5.8 apresenta conceitualmente a técnica PSK.

O método ilustrado acima é denominado 2-PSK porque são utilizadas apenas duas representações de fases diferentes (0° e 180°). A Figura 5.9 mostra outra representação da fase e do bit representado. Um segundo diagrama, denominado **constelação** ou espaço de fase, ilustra o mesmo relacionamento entre as fases e os bits representados.

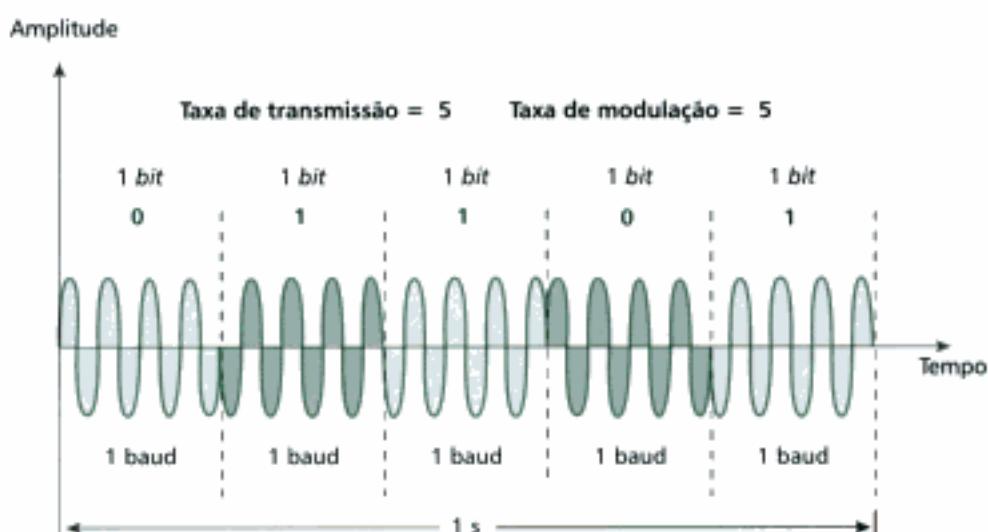


Figura 5.8 PSK.

Bit	Fase
0	0
1	180

Bits



Figura 5.9 Constelação PSK.

A modulação PSK não é suscetível às degradações provocadas por ruídos, que tanto afetam a técnica ASK, ou as limitações de banda da técnica FSK. Isto significa que pequenas variações no sinal podem ser facilmente detectadas por um receptor PSK. Desse modo, em vez de utilizar apenas duas variações de fase em um sinal, cada qual representando apenas um *bit* por vez, podemos utilizar quatro ou mais variações de fase, o que permite representar dois ou mais *bits* ou símbolos por vez (veja Figura 5.10).

A constelação do sinal ilustrado na Figura 5.10 é mostrada na Figura 5.11. Nesse exemplo, uma fase de 0° representa 00; a fase de 90° representa 01; 180° representa 10 e 270° representa 11. Esta técnica é denominada 4-PSK ou Q-PSK. O par de *bits* representado em cada fase é chamado um **dibit**. Podemos transmitir dados com uma eficiência duas vezes maior na técnica 4-PSK em comparação com o método 2-PSK.

Expandindo a idéia para a técnica 8-PSK, em vez de 90° , variamos a fase do sinal em deslocamento regulares de 45° . Usando oito fases diferentes, cada deslocamento é capaz de representar 3 *bits* (**tribit**) por vez. (Como você pode ver, a relação entre o número de *bits* ou símbolos por deslocamento de fase segue a potência de 2. Com quatro fases possíveis, podemos enviar 2 *bits* por vez, isto é, $2^2 = 4$. Já com oito fases, podemos enviar 3 *bits* ao mesmo tempo, pois $2^3 = 8$.) A Figura 5.12 ilustra a relação entre o número de deslocamentos de fase e o valor do *tribit* representado; a técnica 8-PSK é três vezes mais eficiente que a 2-PSK.

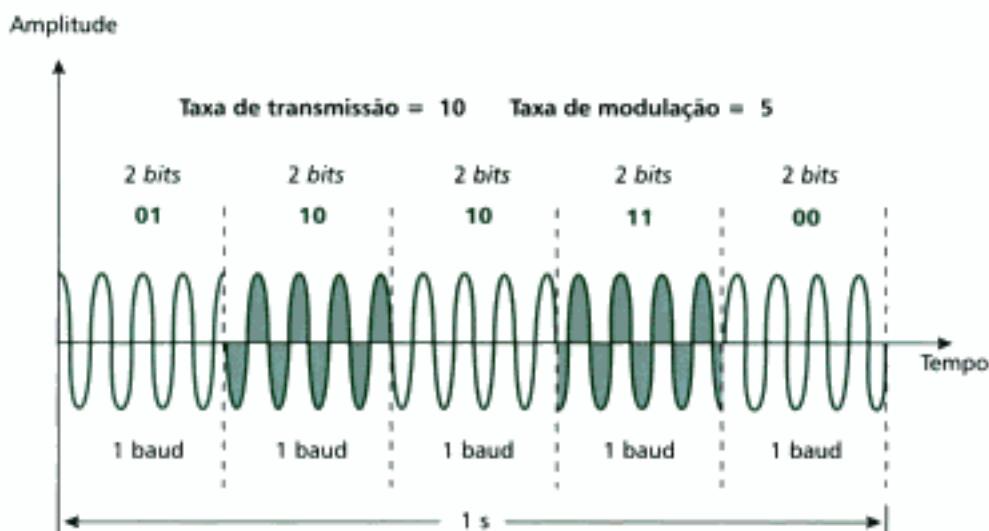


Figura 5.10 O método 4-PSK.

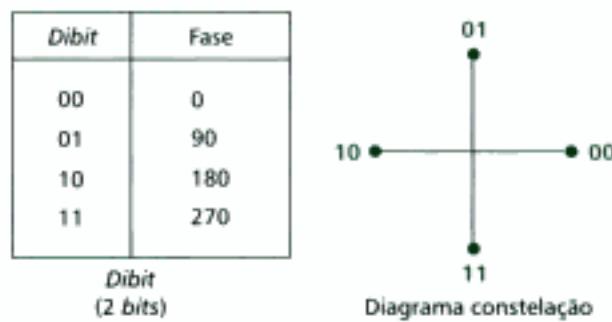


Figura 5.11 Características 4-PSK.

Largura de Banda PSK

A largura de banda mínima requerida na transmissão PSK é a mesma requerida numa transmissão ASK e pelas mesmas razões (veja Figura 5.13). Entretanto, a taxa de transmissão PSK má-

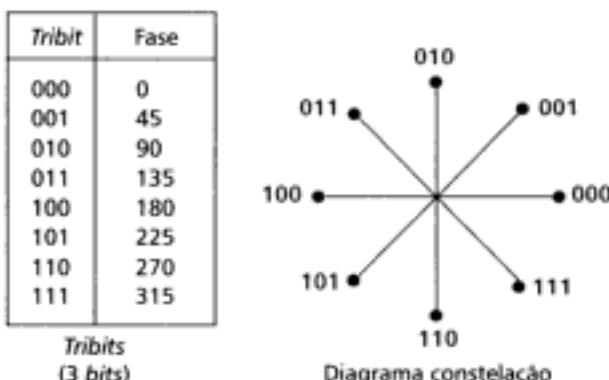


Figura 5.12 Características 8-PSK.

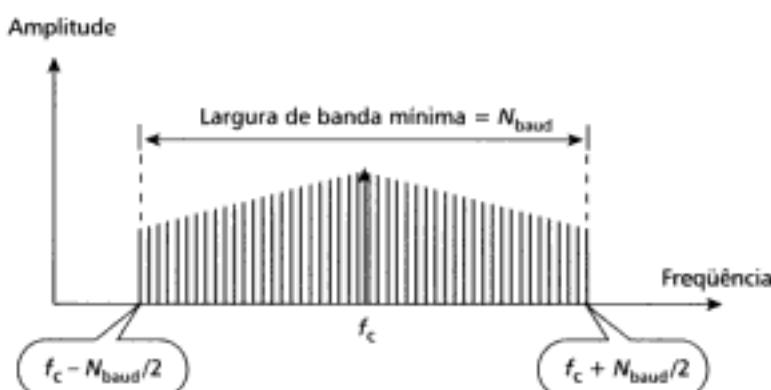


Figura 5.13 Relação entre taxa de modulação e largura de banda PSK.

xima é potencialmente muito superior à taxa de transmissão da técnica ASK. Assim, enquanto as taxas máximas de modulação e de transmissão nas técnicas ASK e PSK se equivalem, a taxa de transmissão na técnica PSK pode ser muito superior (2, 4, 8, etc.) à taxa de modulação, para uma mesma largura de banda.

Exemplo 8

Determine a largura de banda para um sinal 4-PSK transmitido a 2000 bps. Considere que a transmissão ocorre no modo *half-duplex*.

Solução

Para a 4-PSK, a taxa de modulação é numericamente a metade da taxa de transmissão. Assim, a taxa de modulação vale 1000 baud/s. Um sinal PSK requer uma largura de banda igual à taxa de modulação desse sinal. Logo, a largura de banda deve valer 1000 Hz = 1 kHz.

Exemplo 9

Dada uma largura de banda de 5 kHz para uma sinal 8-PSK, quais são as taxas de modulação e de transmissão?

Solução

Para a técnica PSK, a taxa de modulação é numericamente igual à largura de banda. Isto significa que as modulações ocorrem a 5000 baud/s. Mas, a técnica 8-PSK transmite dados a uma taxa de três vezes a taxa de modulação. Portanto, a taxa de transmissão de dados deve ocorrer a 15.000 bps.

Quadrature Amplitude Modulation (QAM)

A modulação PSK está limitada à capacidade do equipamento em distinguir ou detectar pequenas diferenças de fase. Este fator reduz as taxas de transmissão potenciais da técnica PSK.

Até o presente momento, permitimos que fosse alterado apenas uma das três características de um sinal senoidal. Porém, o que acontece se alterarmos duas características simultaneamente? Bom, limitações na largura de banda tornam impraticável combinarmos a técnica FSK com qualquer outro esquema de modulação. Mas, por que não combinar ASK e PSK? Nesse caso, poderíamos através de x variações de fase e de y variações na amplitude resultar em x vezes y símbolos diferentes nos estados de representação, aumentando assim a quantidade de bits representados. A modulação QAM funciona desse modo.

A técnica Quadrature Amplitude Modulation (QAM) é uma combinação das técnicas ASK e PSK elaborada de maneira a aumentar o número de bits transmitidos (bit, díbit, tríbit, etc.) para uma dada taxa de modulação.

São numerosas as possibilidades de implementação da QAM. Teoricamente, um número qualquer de amplitudes pode ser combinado a uma vasta quantidade de variações na fase. A Figura 5.14 ilustra duas configurações possíveis: a 4-QAM e a 8-QAM. Nos dois casos, a quantidade de deslocamentos de amplitude utilizadas é menor que o número de deslocamento de fase. É sempre assim, devido às variações na amplitude serem susceptíveis aos ruídos e requererem grandes diferenças de valor para serem detectadas, o número de fases utilizadas num sistema QAM é sempre

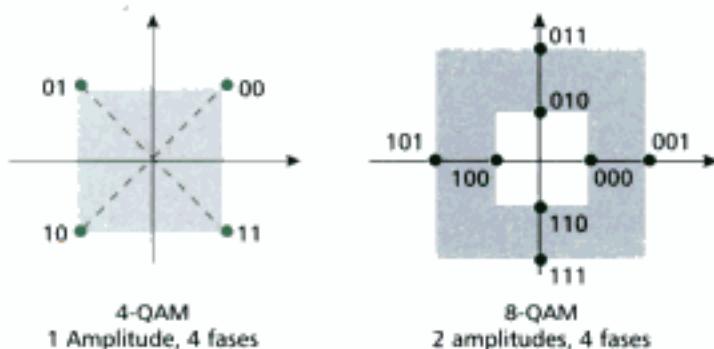


Figura 5.14 A constelação 4-QAM e 8-QAM.

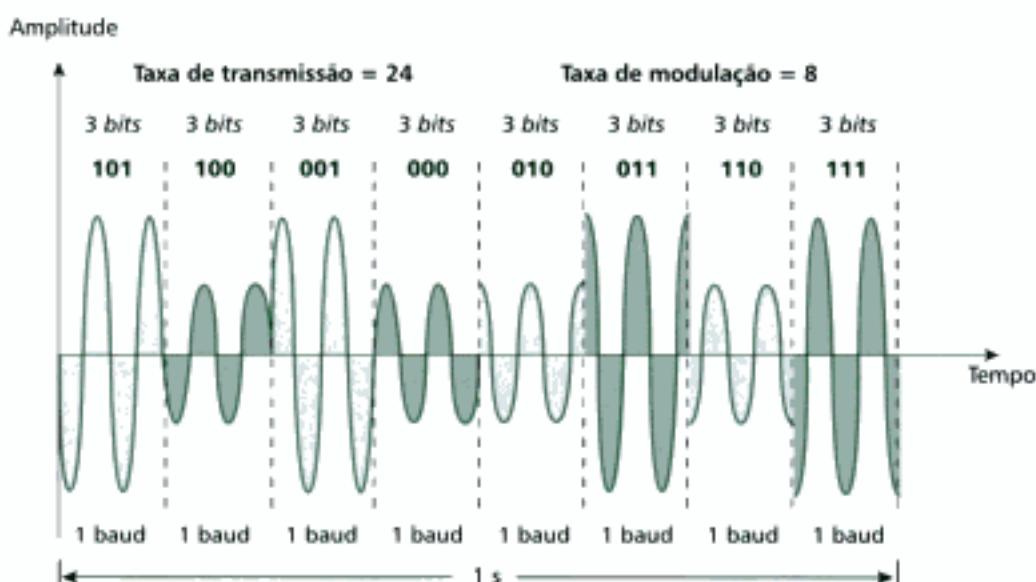


Figura 5.15 Domínio de tempo para o sinal 8-QAM.

superior ao número de amplitudes. Um gráfico desenhado no domínio do tempo correspondente ao sinal 8-QAM da Figura 5.14 é mostrado na Figura 5.15.

Outras configurações geométricas também são possíveis. Três configurações 16-QAM bem conhecidas estão ilustradas na Figura 5.16. No primeiro exemplo, 3 amplitudes e 12 fases, nesse caso, os ruídos podem ser tratados através de mecanismos mais simples visto que a razão entre o número de fases e de amplitudes favorece ao número de fases. Esta é uma recomendação ITU-T. O segundo exemplo, 4 amplitudes e 8 fases, é uma recomendação da OSI. Se você examinar o gráfico cuidadosamente verá que nem toda interseção entre os valores permitidos de fase e de amplitudes é utilizada. De fato, são permitidas 32 variações diferentes. Entretanto, são utilizadas apenas metade dessas combinações, aumentando o grau de mensurabilidade entre os deslocamentos de fase, tornando a legibilidade do sinal no receptor praticamente assegurada. De modo geral, existem muitos outros esquemas de modulação QAM estabelecendo relações estreitas entre a quantidade de amplitudes e fases específicas. Isto significa que os problemas provocados por ruídos associados aos deslocamentos de amplitude podem ser pulverizados através da escolha adequada do esquema QAM. Portanto, a informação contida nos deslocamentos amplitude/fase passa a depender mais da capacidade do receptor em reconhecer os deslocamentos de fase do que os deslocamentos de amplitude. Em geral, podemos dizer que a segunda vantagem da modulação QAM sobre a ASK é a baixa susceptibilidade aos ruídos.

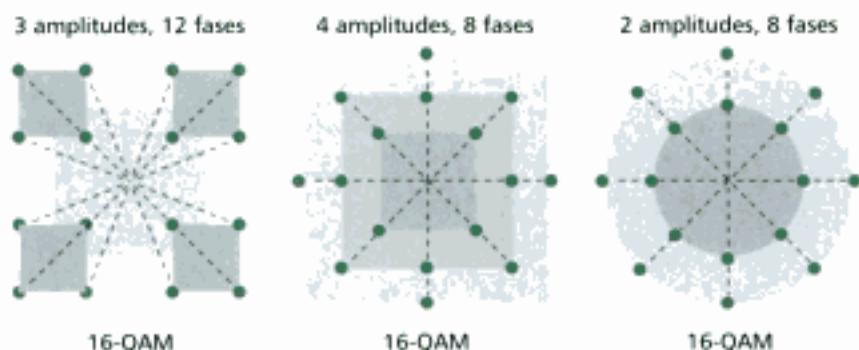


Figura 5.16 Constelação 16-QAM.

Largura de Banda QAM

A transmissão QAM requer uma largura de banda mínima idêntica às transmissões ASK e PSK. A modulação QAM possui as mesmas vantagens que a modulação PSK tem em relação a ASK.

Comparação entre Taxa de Transmissão e Modulação

Consideremos um sinal FSK sendo transmitido a 1200 bps através de uma linha telefônica. Nesse exemplo, a taxa de transmissão é de fato 1200 bps. Cada deslocamento de frequência representa um único *bit* de modo que ela requer 1200 modulações para transmitir 1200 *bits*. Logo, a taxa de modulação é numericamente igual a 1200 baud. Se tomarmos, por exemplo, a modulação 8-QAM são necessárias apenas 400 modulações para transmitir os mesmos 1200 bps. De acordo com a Figura 5.17, um sistema *dibit* possui uma taxa de modulação igual à metade da taxa de transmissão, um sistema *tribit* tem uma taxa de modulação igual a um terço da taxa de transmissão, o sistema **tetrabit** tem uma taxa de modulação de um quarto da taxa de transmissão e assim por diante.

A Tabela 5.1 apresenta uma comparação entre as taxas de transmissão e de modulação para vários métodos de modulação digital.

Exemplo 10

Uma constelação consiste de 8 pontos igualmente espaçados sobre um círculo. Se a taxa de transmissão vale 4800 bps, qual é a taxa de modulação?

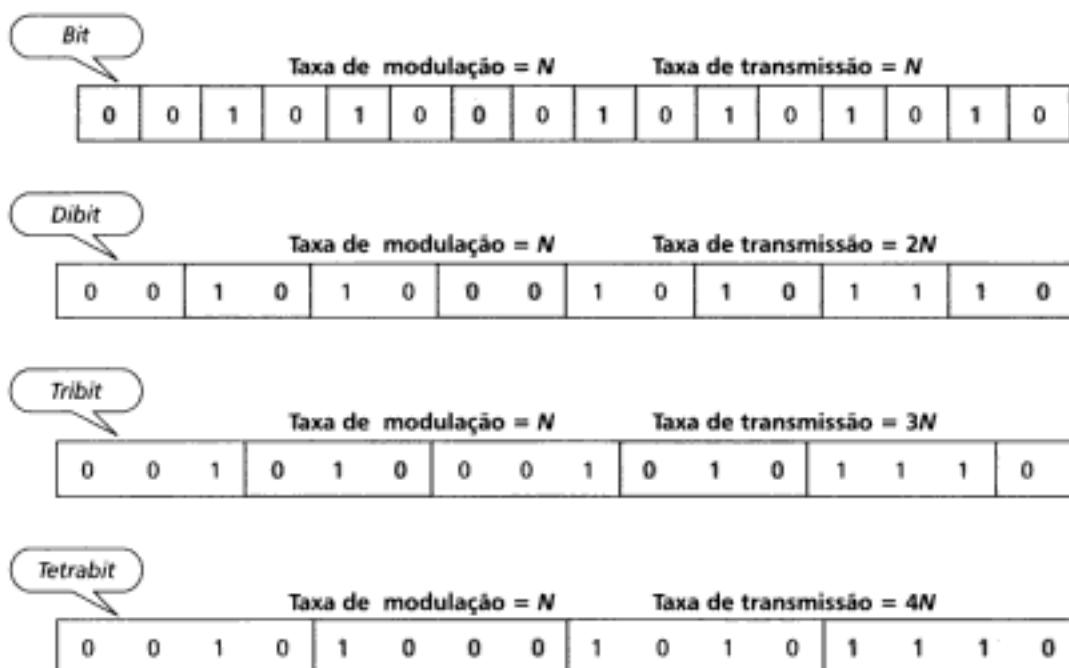


Figura 5.17 Taxa de transmissão e taxa de modulação.

TABELA 5.1 Comparação entre taxa de transmissão e modulação

Modulação	Unidades	Bits/Baud	Taxa de modulação	Taxa de transmissão
ASK, FSK, 2-PSK	Bit	1	N	N
4-PSK, 4-QAM	Dabit	2	N	2N
8-PSK, 8-QAM	Tribit	3	N	3N
16-QAM	Tetrabit	4	N	4N
32-QAM	Pentabit	5	N	5N
64-QAM	Hexabit	6	N	6N
128-QAM	Septabit	7	N	7N
256-QAM	Octabit	8	N	8N

Solução

A constelação sugere uma modulação 8-PSK com pontos igualmente espaçados de 45° entre si. Como $2^3 = 8$, podemos transmitir 3 bits simultaneamente num único intervalo de modulação. Assim, a taxa de modulação deve ser:

$$\frac{4800}{3} = 1600 \text{ baud}$$

Exemplo 11

Determine a taxa de transmissão de um sinal modulado a 1000 baud num sistema 16-QAM.

Solução

Um sinal 16-QAM transmite 4 bits por modulação porque $\log_2 16 = 4$. Portanto,

$$(1000)(4) = 4000 \text{ bps}$$

Exemplo 12

Determine a taxa de modulação de um sinal 64-QAM transmitido a 72.000 bps.

Solução

Um sinal 64-QAM transmite 6 bits por modulação porque $\log_2 64 = 6$. Portanto,

$$\frac{72.000}{6} = 12.000 \text{ baud}$$

5.2 MODEMS ANALÓGICOS

A linha telefônica* comum pode transportar sinais cujas freqüências variam entre 300 e 3300 Hz e, por isso, tem uma largura de banda de 3000 Hz. Toda essa faixa é utilizada para transmissão de voz, onde também coexiste uma grande quantidade de interferências e distorções, e sem que isso provoque perdas de inteligibilidade desse sinal. Contudo, vimos que os sinais digitais requerem um grau de precisão muito maior para assegurar a integridade da transmissão. Isso sugere que a faixa segura para transmissão de dados dentro do canal de voz seja mais estreita dentro do canal de voz. Em geral, para comunicação de dados, utilizamos uma largura de banda do canal menor que a largura de banda dos cabos. Assim, a largura de banda efetiva na transmissão de dados é 2400 Hz, cobrindo uma faixa de freqüências de 600 a 3000 Hz. Hoje em dia algumas linhas telefônicas asseguram uma largura de banda maior que as linhas tradicionais. Contudo, os projetos de novos modems ainda utilizam a capacidade das linhas tradicionais como padrão (veja Figura 5.18).

Um canal de voz possui uma largura de banda de 2400 Hz para transmissão de dados.

Esta largura de banda define a natureza da banda básica, a qual necessitaremos para efetivar as modulações se quisermos usar a banda do canal de voz para transmissão de dados. Os dispositivos que possibilitam a transmissão de dados através de um canal de voz são denominados modems.

O acrônimo **MODEM** é uma palavra composta que significa MODulador/DEModulador. Um **modulador** gera um sinal analógico tipo canal passa-banda na saída a partir dos dados binários de entrada. Um **demodulador** recupera os dados binários na saída a partir do sinal modulado na entrada.

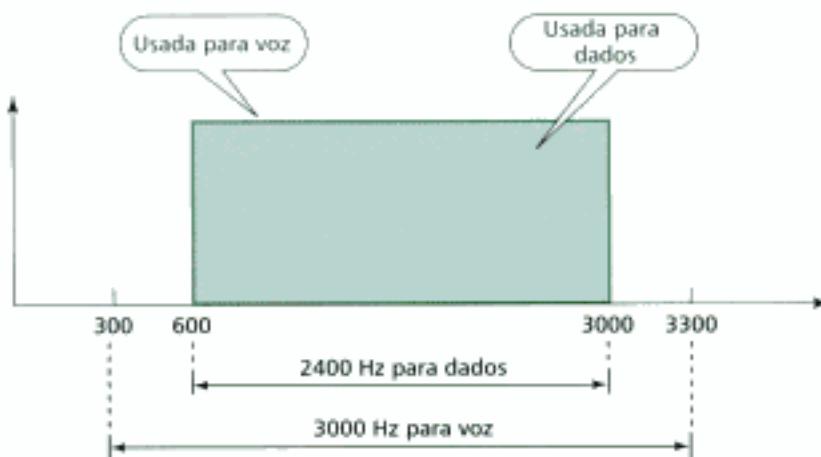


Figura 5.18 Largura de banda da linha telefônica.

* N. de R. T.: A linha telefônica é denominada tradicionalmente canal de voz e, no Brasil, situa-se entre 300 e 3400 Hz, perfazendo uma largura de banda de 3100 Hz.

Modem significa modulator/demodulator.

A Figura 5.19 ilustra a posição de dois modems dentro de um sistema básico de comunicação de dados. O computador à esquerda envia dados binários para a porção moduladora do modem. Os dados são enviados com um sinal analógico dentro do canal de voz. O modem à direita recebe o sinal analógico, demodula através da sua porção demoduladora e entrega os dados ao computador remoto à direita. A comunicação pode ser bidirecional de modo que o computador à direita pode retornar a comunicação de dados ao computador à esquerda através do mesmo mecanismo de modulação/demodulação.

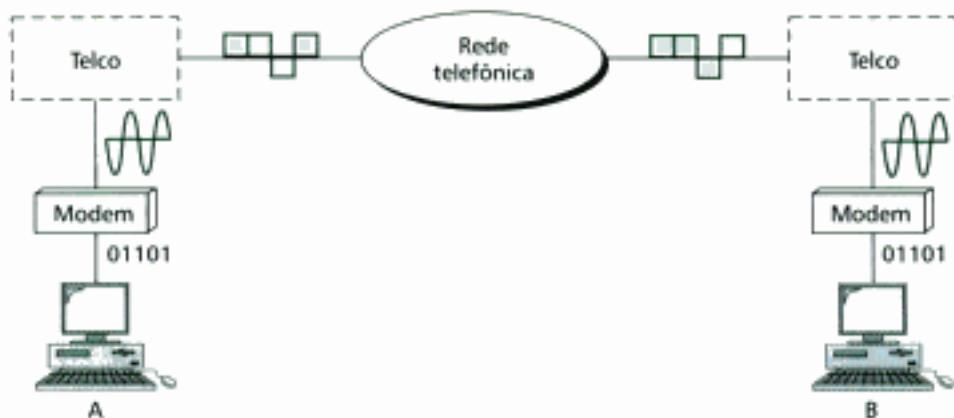


Figura 5.19 Modulação/demodulação.

Padrões de Modem

Hoje em dia, boa parte dos modems populares estão baseados nos padrões da **série-V** publicado pela ITU-T. Discutiremos algumas das séries mais recentes.

V.32

Os modems baseados na série V.32 combinam técnicas de modulação e de codificação denominada **Modulação com Codificação Treliça** ou **TCM (Trellis Coded Modulation)**. O esquema de treliça é essencialmente QAM mais um *bit* de redundância. O fluxo de dados é dividido em um padrão de 4 *bits*. Porém, em vez de transmitir *tetrabits* são enviados pentabits (padrão de 5 *bits*). O valor do *bit* extra de redundância é calculado a partir dos valores dos quatro *bits* de dados.

Em qualquer sistema de QAM, o receptor compara cada sinal recebido com o mapa de todos os pontos válidos dentro da constelação de pontos e seleciona o ponto mais próximo do valor pretendido. Um sinal distorcido por ruídos durante uma transmissão pode modificar o valor desejado para o ponto dentro da constelação de forma que o sistema receptor aponte algum outro ponto adjacente dentro da constelação como resultado da demodulação. Isto resulta num erro da transmissão de dados e na perda da identificação do ponto dentro da constelação. Quanto mais próximos estiverem os pontos dentro da constelação mais facilmente podem ocorrer estes tipos de erros e perdas de identificação. Por isso, um *bit* redundante é adicionado ao padrão *tetrabit* para identificá-lo univocamente e, assim, reduzir a quantidade de erros possíveis. Por esse motivo, um sinal com codificação treliça é muito mais útil e apresenta muito menos problemas de leitura quando distorcido por ruídos, se comparado à técnica QAM padrão.

A série V.32 utiliza técnica 32-QAM modulando a uma taxa de 2400 baud. Como somente 4 dos 5 *bits* representam dados, o resultado é uma velocidade de transmissão de $4 \times 2400 = 9600$ bps. A constelação de pontos e a largura de banda são mostrados na Figura 5.20.

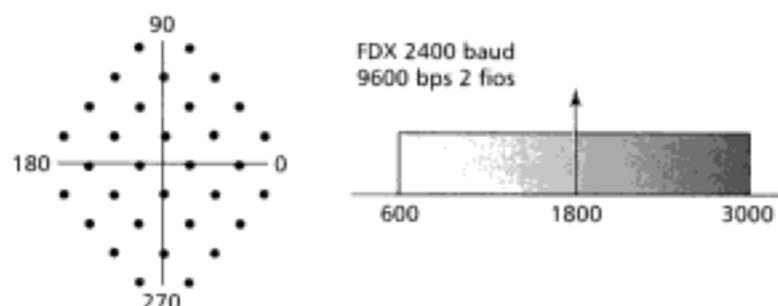


Figura 5.20 A constelação V.32 e a largura de banda.

V.32bis

Os modems baseados na série **V.32bis** foram padronizados inicialmente pela ITU-T de modo a suportar transmissões até 14.400 bps. Os V.32bis utilizam transmissão 128-QAM (7 bits/baud com um 1 bit para controle de erro) a uma taxa de modulação de 2400 baud ($2400 \times 6 = 14.400$ bps).

Dentre os melhoramentos introduzidos pela série V.32bis, estão incluídos os recursos de *fall-back* e *fall-forward* automáticos que habilitam o modem a ajustar a velocidade da transmissão/recepção de acordo com o modem remoto e dependendo da qualidade da linha ou do sinal. A constelação e a largura de banda estão mostradas na Figura 5.21.

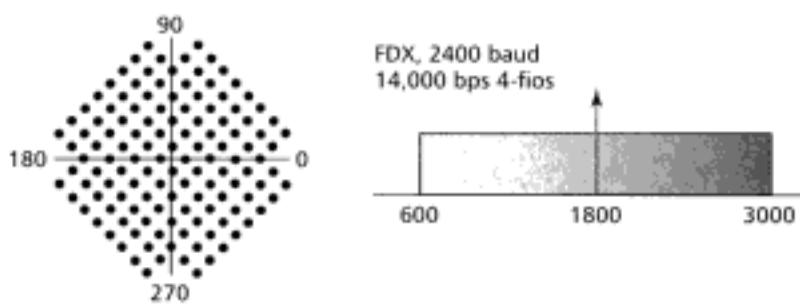


Figura 5.21 A constelação V.32bis e a largura de banda.

V.34bis

Os modems da série V.34bis proporcionam taxas de transmissão de 28.800 bps na constelação com 960 pontos e até 33.600 bps na constelação com 1664 pontos.

V.90

Os modems tradicionais possuem um limite teórico máximo de transferência de dados da ordem de 33,6 kbps determinado através da lei de Shannon (veja Capítulo 3). Entretanto, os modems da série V.90 podem atingir velocidades de transmissão de até 56.000 bps e por isso são denominados **modems de 56K**. Estes modems são recomendados somente quando as linhas telefônicas apresentarem-se digitalizadas através de centrais telefônicas digitais. Geralmente esses modems funcionam assimetricamente, isto é, a taxa de transferência é diferente nos dois sentidos de tráfego. Por exemplo, a taxa de transferência durante um *downloading* atinge um máximo de 56 Kbps, enquanto que durante um *uploading* (fluxo de dados do PC para o provedor de Internet, por exemplo) chega a no máximo 33,6 kbps. Então, esses modems violam a lei de Shannon? Definitivamente não. O princípio de funcionamento em que se baseiam é diferente. Passemos a uma comparação das duas técnicas.

Modems Tradicionais Nestes tipos de modems, a troca de dados entre dois computadores (A e B), através da linha telefônica digital, acontece de acordo com a Figura 5.22.

Após passar pelo processo de modulação, o sinal analógico alcança a central de comutação telefônica da empresa de telefonia local onde ele é amostrado e digitalizado para então ser coloca-

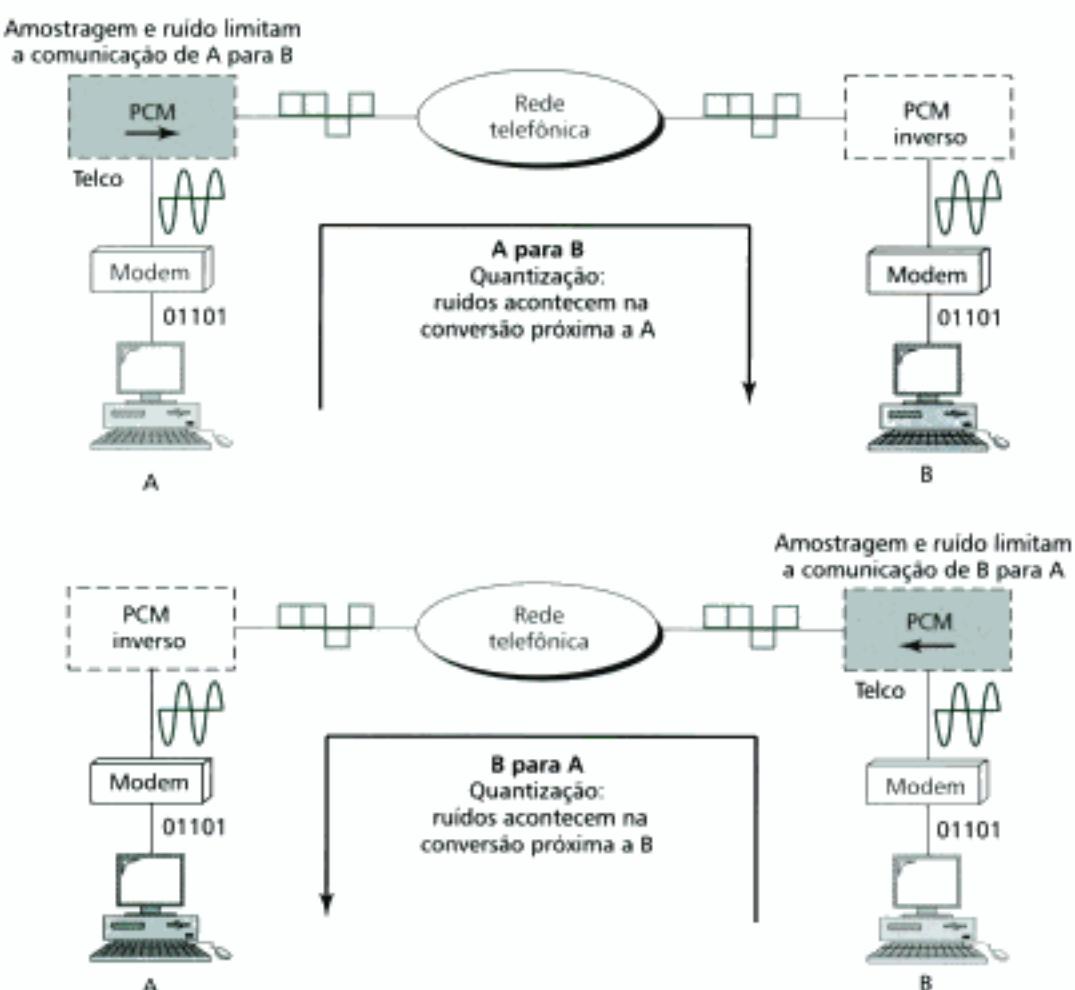


Figura 5.22 Modem convencional.

do na linha digital. O ruído de quantização introduzido no sinal no estágio de amostragem limita a taxa de transmissão conforme estabelece a lei de Shannon. Este limite é de 33,6 kbps.

Como a amostragem é feita nas duas extremidades da linha, a taxa de transmissão máxima nesses meios é de 33,6 kbps.

Modems 56K Boa parte das nossas comunicações nos dias de hoje ocorrem através da Internet. Ainda utilizamos modems para fazer *uploads* dos dados para Internet e *downloads* dos dados da Internet (veja Figura 5.23).

Durante uma operação de **uploading** o sinal analógico ainda deve ser amostrado na central de comutação, o que significa que a taxa de transferência nos *uploadings* está limitada a 33,6 kbps. Entretanto, não é feita nenhuma amostragem durante a operação de **downloading**. O sinal não é afetado pelo ruído de quantização e não fica sujeito à limitação imposta pela lei de Shannon. A taxa de transferência máxima na direção do *uploading* é de 33,6 kbps, mas a taxa de transferência na direção do *downloading* passa a ser 56 kbps.

Podemos entender o porquê do limite de 56 kbps. As companhias telefônicas amostram os sinais a 8000 vezes por segundo com 8 bits por amostra. Um dos bits em cada amostra é utilizado para os propósitos de controle, o que significa que cada amostra corresponde a 7-bits. A taxa é assim 8000×7 ou 56.000 bps ou 56 kbps.

V.92

O padrão imediatamente acima da série V.90 é a **V.92**. Estes modems conseguem ajustar as velocidades deles e, se o nível de ruído na linha permitir, chegar a taxas de *uploads* de 48 kbps. A taxa de *downloading* continua a ser 56 kbps. Estes modems possuem algumas características adicionais:

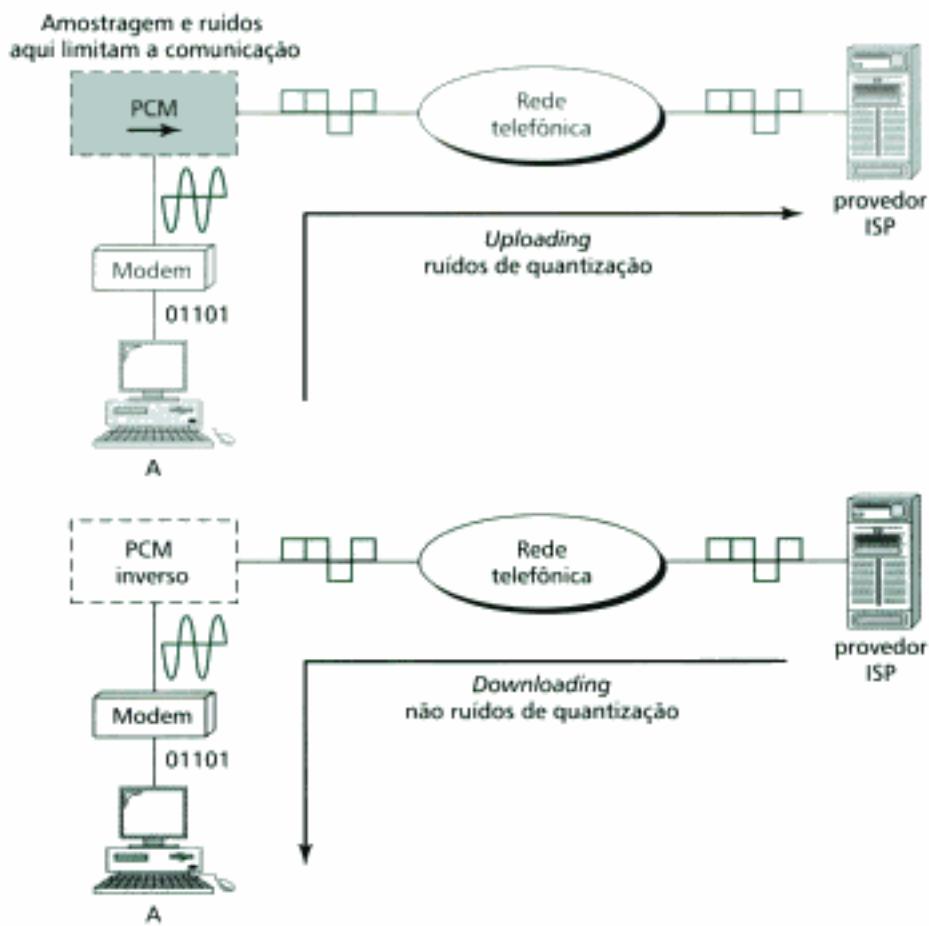


Figura 5.23 Modem 56K.

nais. Por exemplo, se a linha possui o serviço de chamada em espera, um modem dessa série consegue interromper a conexão de Internet quando receber uma chamada.

5.3 MODULAÇÃO DE SINAIS ANALÓGICOS

A modulação de um sinal analógico ou conversão de analógico para analógico é a representação da informação analógica através de um sinal analógico. Às vezes somos questionados porquê precisamos modular um sinal analógico, já que ele se encontra originalmente na forma analógica. A modulação é necessária se o meio é de natureza passa-banda ou se dispusermos somente de uma largura de banda para efetuar nossas transmissões. Um excelente exemplo é o rádio. Algum órgão competente ligado ao governo atribui uma banda para cada estação de rádio. O sinal analógico produzido em cada estação é um sinal tipicamente passa-baixas e encontra-se na mesma faixa das demais estações. Para que sejamos capazes de ouvir estações diferentes esses sinais passa-baixas devem ser deslocados, cada qual para uma faixa diferente.

A Figura 5.24 mostra o relacionamento entre a informação analógica de entrada, o dispositivo de hardware que efetua a modulação e o sinal analógico de saída.

A **modulação analógica** pode ser realizada de três formas: **Amplitude Modulation (AM)**, **Frequency Modulation (FM)** e **Phase Modulation (PM)***. Observe a Figura 5.25.

* N. de R. T.: Deixaremos os métodos de modulação analógicos em inglês. Entretanto, as melhores traduções são: AM: Modulação em amplitude; FM: Modulação em frequência; PM: Modulação em fase.



Figura 5.24 Modulação de analógico para analógico.

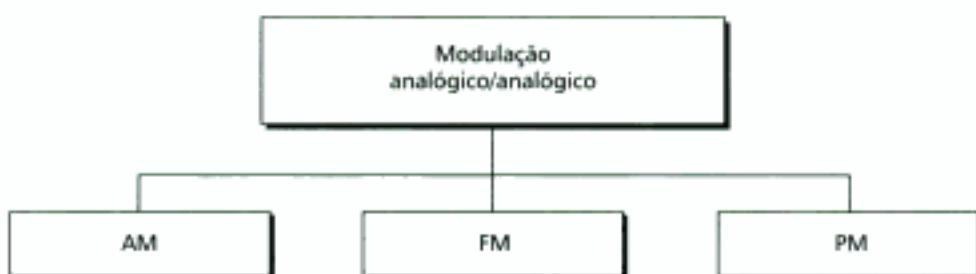


Figura 5.25 Tipos de modulação analógico para analógico.

Amplitude Modulation – AM

Numa transmissão em AM, o sinal da portadora é modulado de modo que sua amplitude varie de acordo com as variações da amplitude do sinal modulante. A freqüência e a fase da portadora permanecem as mesmas; as variações ocorrem somente na amplitude da portadora, seguindo as variações do sinal de informação (sinal modulante). A Figura 5.26 mostra como funciona esse conceito. O sinal modulante é na verdade um envelope da portadora.

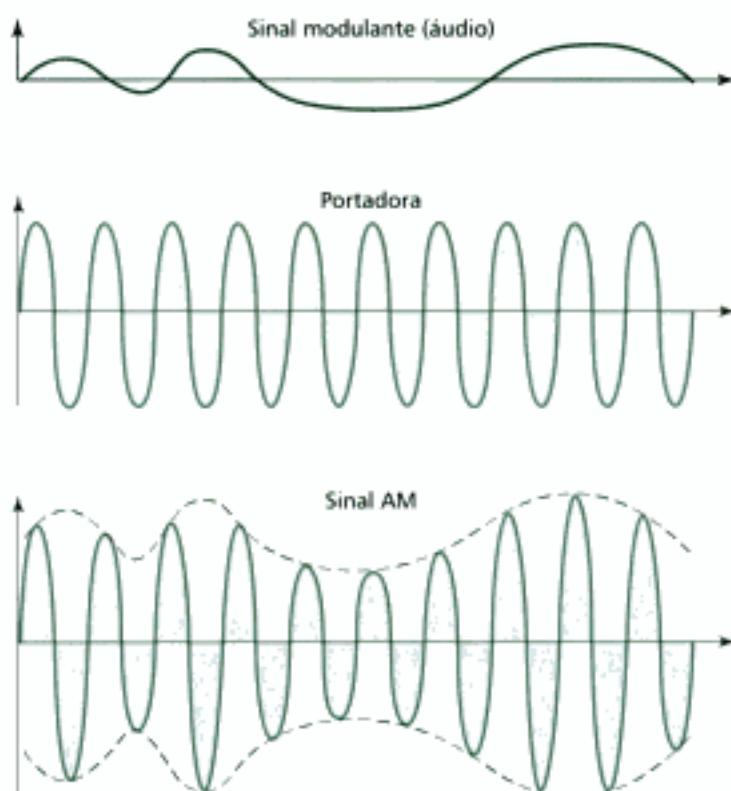


Figura 5.26 Modulação em amplitude (AM).

Largura de Banda AM

A largura de banda de um sinal AM é igual a duas vezes a largura de banda do sinal modulante e deve cobrir uma faixa centrada na freqüência da portadora (veja Figura 5.27). A porção sombreada do gráfico é o espectro de freqüência do sinal.

A largura de banda de um sinal de áudio (voz e música) é tipicamente 5 kHz. Desse modo, uma estação de rádio necessita de uma largura de banda mínima de 10 kHz. De fato, a Federal Communications Commission (FCC) permite que cada estação de rádio AM ocupe uma banda de 10 kHz.

As estações de rádio AM estão distribuídas numa faixa que se estende de 530 a 1.700 kHz (1.7 MHz). Todavia, a freqüência da portadora de cada estação deve estar separada de 10 kHz das portadoras adjacentes para evitar interferência de uma estação noutra. Se a freqüência de uma portadora de AM for 1100 kHz, a freqüência da portadora da próxima estação não pode ser menor que 1110 kHz (veja Figura 5.28).

A largura de banda total requerida para transmissão AM pode ser determinada a partir da largura de banda do sinal de áudio: $BW_t = 2 \times BW_m$.

BW_m = largura de banda do sinal modulante (áudio)

BW_t = largura de banda total

f_c = freqüência da portadora

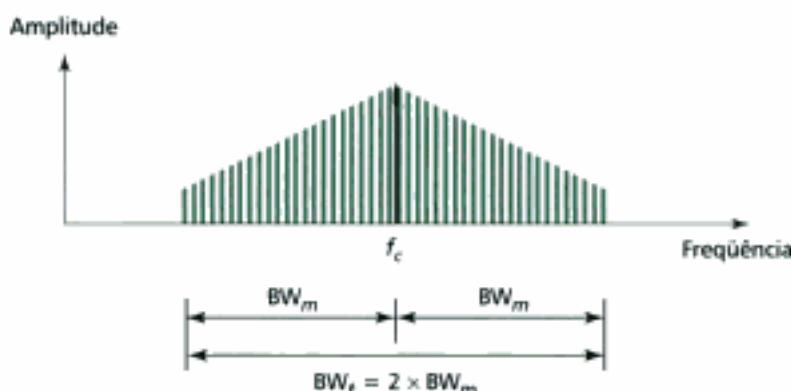


Figura 5.27 Largura de banda AM.



Figura 5.28 Alocação de banda AM.

Exemplo 13

Temos um sinal de áudio com uma largura de banda de 4 kHz. Qual deve ser a largura de banda necessária se desejarmos modular esse sinal usando AM? Ignore a regulamentação do FCC.

Solução

Um sinal de AM requer um sinal de largura de banda igual a duas vezes a frequência do sinal original:

$$BW = 2 \times 4 \text{ kHz} = 8 \text{ kHz}$$

Frequency Modulation – FM

Na transmissão em FM, a frequência do sinal da portadora é modulada de modo a seguir as variações dos níveis de tensão do sinal modulante. O valor de pico e a fase do sinal da portadora permanecem constantes, mas como a amplitude do sinal modulante varia, a frequência do sinal da portadora varia em resposta à essa variação. A Figura 5.29 ilustra a relação existente entre o sinal modulante, o sinal da portadora e o sinal FM resultante.

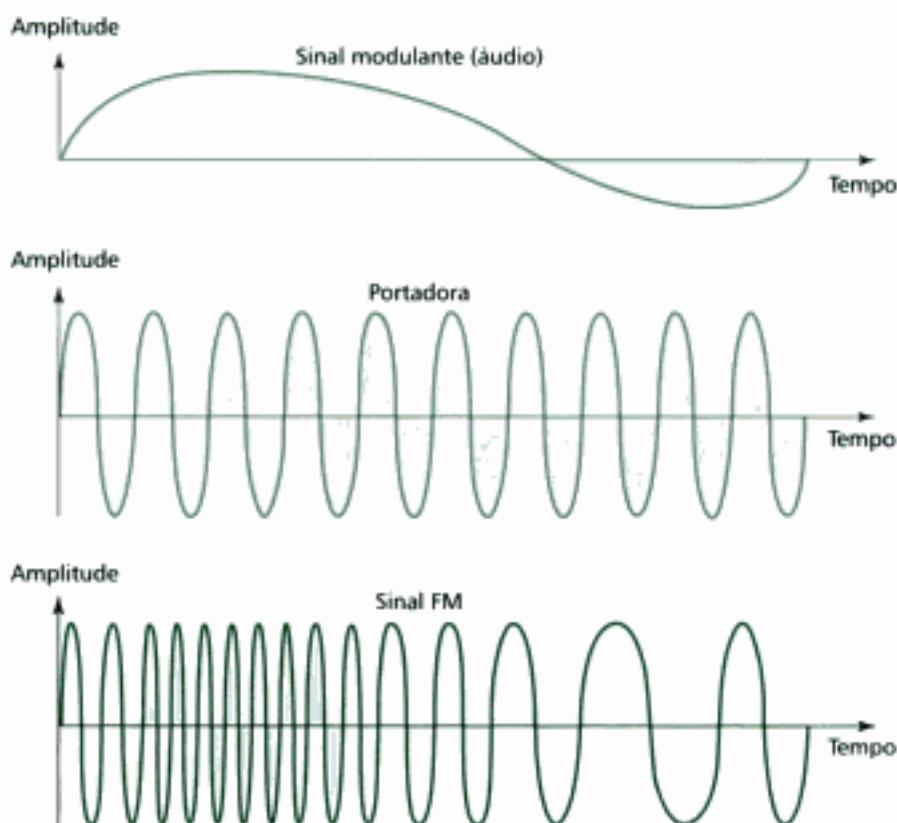


Figura 5.29 Modulação em freqüência.

Largura de Banda FM

A largura de banda de um sinal FM é igual a 10 vezes a largura de banda do sinal modulante e, como na modulação AM, deve cobrir uma faixa centrada na freqüência da portadora. A Figura 5.30 mostra tanto a largura de banda como o espectro de freqüência sombreado de um sinal de FM.

A largura de banda de um sinal de áudio (voz e música) de radiodifusão em estéreo é aproximadamente 15 kHz. Assim, cada estação de FM necessita de uma largura mínima de 150 kHz. O FCC concede 200 kHz (0.2 MHz) para cada estação com uma pequena tolerância (janela) para salvaguardar as bandas de estações muito próximas.

As estações de FM estão todas distribuídas numa faixa de freqüências entre 88 e 108 MHz. Essas estações estão separadas de 0,2 MHz entre si para evitar que as bandas delas se sobreponham, provocando interferência mútua. Para aumentar ainda mais a privacidade de cada estação,

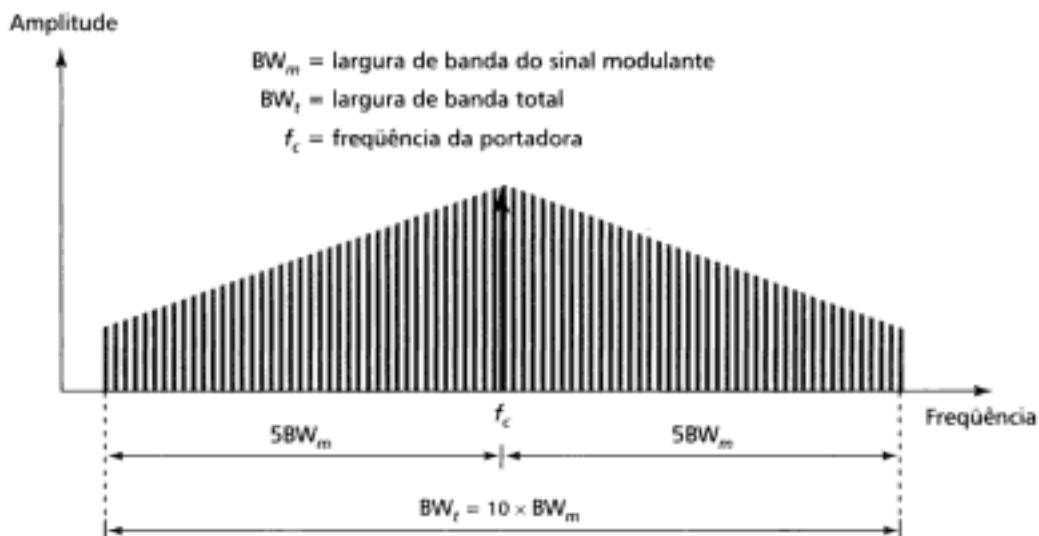


Figura 5.30 Largura de banda em FM.

A largura de banda total requerida para transmissão AM pode ser determinada a partir da largura de banda do sinal de áudio: $BW_t = 10 \times BW_m$.

o FCC requer que, numa dada área, somente as bandas alternadas podem ser utilizadas. As outras permanecem livres (inutilizadas) para prevenir qualquer possibilidade de duas estações se interferirem. Na faixa de 88 a 108 MHz temos potencialmente 100 estações de FM numa determinada área, das quais apenas 50 podem operar livremente. A Figura 5.31 ilustra esse conceito.

A largura de banda de um sinal de áudio estereofônico é tipicamente 15kHz. Por essa razão, uma estação FM necessita de uma largura de banda mínima de 150kHz (dez vezes maior). Por segurança, para evitar interferência entre as estações, o FCC recomenda uma largura de banda mínima de 200kHz (0,2MHz).

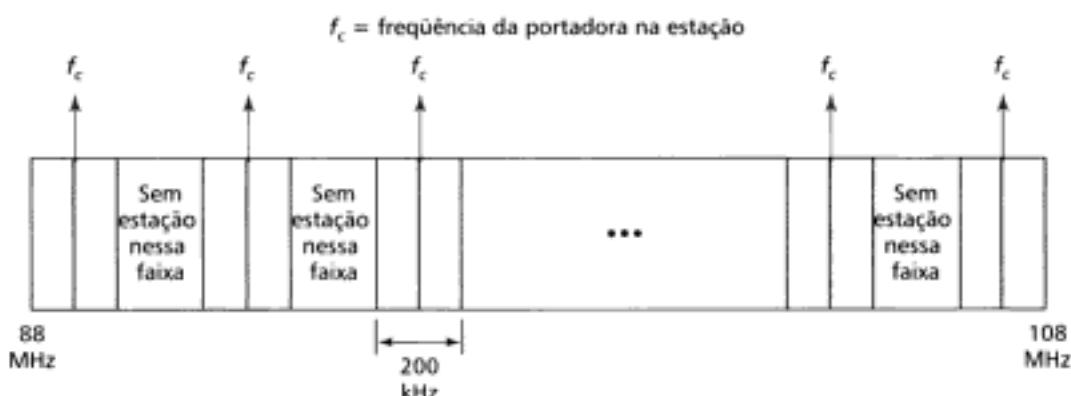


Figura 5.31 Modulação em freqüência.

Exemplo 14

Se tivermos um sinal de áudio com uma largura de banda de 4 kHz, qual é a largura de banda necessária para transmiti-lo se desejarmos modular esse sinal em FM? Ignore a regulamentação do FCC.

Solução

Um sinal de FM requer uma largura de banda 10 vezes maior que o sinal original. Assim:

$$BW = 10 \times 4 \text{ kHz} = 40 \text{ kHz}$$

Phase Modulation – PM

Devido à simplicidade do *hardware* requerido para realizá-la, a modulação PM é utilizada em alguns sistemas como uma alternativa para a modulação em freqüência. Numa transmissão PM, a fase do sinal da portadora é modulada de maneira a seguir as variações do nível de tensão do sinal modulante. O valor de pico e a freqüência do sinal da portadora permanecem inalterados, mas como a amplitude do sinal de informação varia, a fase do sinal da portadora varia em resposta. A análise e o resultado final (sinal modulado) são similares àqueles mostrados na modulação em freqüência.

5.4 TERMOS-CHAVE

Amplitude Shift Keying (ASK)	Modulador
Constelação	Phase Shift Keying (PSK)
Demodulação	Quadrature Amplitude Modulation (QAM)
Demodulador	Série-V
<i>Díbit</i>	Sinal portador
<i>Downloading</i>	Taxa de modulação
Frequency Shift Keying (FSK)	Taxa de transmissão
Modem	Tetrabit
Modem 56k	Tribit
Modulação	Uploading
Modulação analógica	V.32
Modulação com Codificação Treliça	V.32bis
Modulação digital	V.34bis
Modulação em Amplitude (AM)	V.90
Modulação em Fase (PM)	V.92
Modulação em Freqüência (FM)	

5.5 RESUMO

- A modulação digital pode ser implementada através dos seguintes métodos:
 Amplitude Shift Keying (ASK) – varia a amplitude do sinal da portadora.
 Frequency Shift Keying (FSK) – varia a freqüência do sinal da portadora.
 Phase Shift Keying (PSK) – varia a fase do sinal da portadora.
 Quadrature Amplitude Modulation (QAM) – varia tanto a fase quanto a freqüência do sinal da portadora.
- QAM permite transmitir dados nas taxas mais elevadas entre os métodos de transmissão digital.
- Taxa de transmissão (*bit rate*) e taxa de modulação (*baud rate*) não são sinônimos. *Bit rate* é o número de *bits* transmitidos por segundo. *Baud rate* é o número de modulações (sinalizações) transmitidas por segundo. Num sinal de modulação podem estar representados um ou mais *bits*.
- A largura de banda mínima requerida pelas modulações ASK e PSK é numericamente igual à taxa de modulação do canal.
- A largura de banda mínima (BW) requerida pela modulação FSK é: $BW = N_{baud} + f_{c1} - f_{c0}$, onde N_{baud} é a taxa de modulação, f_{c1} é a freqüência que representa o *bit* 1 e f_{c0} é a freqüência que representa o *bit* 0.
- Uma linha telefônica tradicional usa freqüências compreendidas entre 600 e 3000 Hz para transmissão de dados.
- A modulação ASK é especialmente suscetível a ruídos.
- A modulação FSK utiliza duas portadoras, por isso requer uma largura de banda maior que as modulações ASK e PSK.
- As modulações PSK e QAM têm duas vantagens sobre a modulação ASK:
 São menos susceptíveis a ruídos
 Cada variação no sinal pode representar mais de um *bit*.
- A codificação treliça é uma técnica que usa o conceito de redundância para prover um mecanismo de verificação de erros.

- Os modems de 56k são assimétricos, isto é, eles proporcionam *downloads* a 56 kbps e permitem *uploads* a 33,6 kbps.
- A modulação analógica pode ser implementada através dos seguintes métodos:
 - Modulação em Amplitude (Amplitude Modulation – AM)
 - Modulação em Freqüência (Frequency Modulation – FM)

- Modulação em Fase (Phase Modulation – PM)
- Em um rádio AM, a largura de banda do sinal modulado deve ser duas vezes a largura de banda do sinal modulante.
- Em um rádio FM, a largura de banda do sinal modulado deve ser dez vezes a largura de banda do sinal modulante.

5.6 PRATIQUE OS CONHECIMENTOS ADQUIRIDOS

Questões de Revisão

1. O que é modulação digital?
2. O que é modulação analógica?
3. Por que a modulação em freqüência é superior à modulação em amplitude?
4. Qual é a maior vantagem da técnica QAM sobre as técnicas ASK e PSK?
5. Quais são os métodos de conversão de um sinal digital em um sinal analógico?
6. Qual é a diferença entre taxa de transmissão (*bit rate*) e taxa de modulação (*baud rate*)? Cite dois exemplos, um onde as duas taxas são iguais e outro onde as duas taxas são diferentes.
7. O que significa modulação?
8. Qual é o propósito do sinal da portadora numa modulação?
9. De que forma a taxa de modulação está relacionada à largura de banda de transmissão em ASK?
10. De que forma a taxa de modulação está relacionada à largura de banda de transmissão em FSK?
11. De que forma a taxa de modulação está relacionada à largura de banda de transmissão em PSK?
12. Que tipo de informação podemos retirar de uma constelação de pontos de uma modulação?
13. De que forma a taxa de modulação está relacionada à largura de banda de transmissão em QAM?
14. De que forma a QAM está relacionada a ASK e PSK?
15. O que torna a modulação PSK superior a ASK?
16. O que significa o acrônimo *modem*?
17. Qual é a função de um modulador? Qual é a função de um demodulador?
18. Explique a assimetria nos modems de 56k.
19. Por que necessitamos de modems nas comunicações telefônicas?
20. A largura de banda mínima de um sinal ASK poderia ser igual à taxa de transmissão. Explique porque isso é impossível para FSK.
21. Como a modulação AM difere da ASK?
22. Como a modulação FM difere da FSK?
23. Compare, em termos do sinal modulante, as larguras de banda FM e AM.

Questões de Múltipla Escolha

24. ASK, PSK, FSK e QAM são exemplos de modulação _____.
 a. Digital para digital
 b. Digital para analógico
 c. Analógico para analógico
 d. Analógico para digital
25. AM e FM são exemplos de modulação _____.
 a. Digital para digital
 b. Digital para analógico
 c. Analógico para analógico
 d. Analógico para digital
26. Na modulação QAM tanto a fase quanto a _____ do sinal da portadora são variadas.
 a. Amplitude
 b. Freqüência
 c. Taxa de transferência
 d. Taxa de modulação
27. Qual das seguintes técnicas é a mais afetada pelo ruído?
 a. PSK
 b. ASK
 c. FSK
 d. QAM

28. Se a taxa de modulação de um sinal 4-PSK é 400 baud, a taxa de transmissão vale _____ bps.
- 100
 - 400
 - 800
 - 1600
29. Se a taxa de transmissão para um sinal ASK é 1200 bps, a taxa de modulação correspondente vale _____ baud.
- 300
 - 400
 - 600
 - 1200
30. Se a taxa de transmissão para um sinal FSK é 1200 bps, a taxa de modulação correspondente vale _____ baud.
- 300
 - 400
 - 600
 - 1200
31. Se a taxa de transmissão de um sinal QAM é 3000 bps e numa modulação são representados 3 bits (*tribit*), qual é a taxa de modulação (em baud) desse sinal?
- 300
 - 400
 - 1000
 - 1200
32. Se a taxa de modulação de um sinal QAM é 3000 baud e numa modulação são representados 3 bits (*tribit*), qual é a taxa de transmissão (em bps) desse sinal?
- 300
 - 400
 - 1000
 - 9000
33. Se a taxa de modulação para um sinal QAM vale 1800 baud e a taxa de transmissão vale 9000 bps, quantos bits são transmitidos por intervalo de modulação?
- 3
 - 4
 - 5
 - 6
34. Num sinal 16-QAM, há 16 _____.
 a. Símbolos
 b. Amplitudes
 c. Fases
 d. bps
35. Que técnica de modulação manipula *tribits*, oito deslocamentos de fases diferentes e uma de amplitude?
- FSK
 - 8-PSK
 - ASK
 - 4-PSK
36. Dado um sinal de rádio AM com uma largura de banda de 10 kHz e de componente de mais alta freqüência em 705 kHz, qual é a freqüência da portadora?
- 700 kHz
 - 705 kHz
 - 710 kHz
 - Não pode ser determinada a partir da informação do enunciado.
37. Um sinal modulado é formado por _____.
- Variações do sinal modulante provocadas pela portadora.
 - Variações na portadora provocadas pelo sinal modulante.
 - Quantização da fonte de dados.
 - Amostragens para a freqüência de Nyquist.
38. Se as regulamentações do FCC são seguidas, as freqüências das portadoras adjacentes são separadas de _____.
- 5 kHz
 - 10 kHz
 - 200 kHz
 - 530 kHz
39. Se as regulamentações do FCC são seguidas, existem potencialmente _____ estações FM teóricas para uma determinada área.
- 50
 - 100
 - 133
 - 150
40. Se um sinal ASK for decomposto, o resultado será _____.
- Sempre uma onda senoidal.
 - Sempre duas ondas senoidais.
 - Um número infinito de ondas senoidais.
 - Nenhuma das respostas anteriores.
41. A largura de banda de um sinal de FM requer 10 vezes a banda do sinal _____.
- Portador
 - Modulante
 - Bipolar
 - Amostrado

42. A modulação de um sinal analógico pode ser implementada através da variação da _____ do sinal da portadora.
- Amplitude
 - Freqüência
 - Fase
 - Qualquer uma das opções acima.
43. Para uma linha telefônica, a largura de banda do sinal de voz é usualmente _____ largura de banda de transmissão de dados.
- Equivalente a
 - Menor que a
 - Maior que a
 - Duas vezes a
44. Para a uma dada taxa de transmissão, a largura de banda mínima do sinal ASK é _____ a largura de banda mínima do sinal FSK.
- Equivalente
 - Menor que
 - Maior que
 - Duas vezes
45. Quando a taxa de transmissão de um sinal FSK aumenta, a largura de banda _____.
- Diminui
 - Aumenta
 - Permanece a mesma
 - Dobra
46. Para FSK, quando a diferença entre as freqüências de duas portadoras aumenta, a largura de banda _____.
- Diminui
 - Aumenta
 - Permanece a mesma
 - Cai pela metade
47. Que padrão ITUT para modem utiliza a codificação *trellis* ou de treliça?
- V.32
 - V.33
 - V.34
 - (a) e (b)
48. Na codificação *trellis* o número de bits de dados é _____ número de bits transmitidos.
- Igual ao
 - Menor que o
 - Maior que o
 - O dobro do
49. Qual é o segredo da codificação *trellis*?
- Estreitar a banda do sinal.
 - Simplificar a modulação.
 - Aumentar a taxa de transmissão de dados.
 - Reducir a taxa de erros.
50. As taxas de transmissão e de modulação são iguais em qual tipo de sinal?
- FSK
 - QAM
 - 4-PSK
 - Todas anteriores
51. Um modulador converte - um sinal _____ num sinal _____.
- Digital; analógico
 - Analógico; digital
 - PSK; FSK
 - FSK; PSK
52. Um modem de 56k pode fazer *download* a uma taxa de _____ kbps e *upload* a uma taxa de _____ kbps.
- 33,6; 33,6
 - 33,6; 56,6
 - 56,6; 33,6
 - 56,6; 56,6

Exercícios

53. Determine a taxa de modulação para cada taxa de transmissão e tipo de modulação:
- 2000 bps, FSK
 - 4000 bps, ASK
 - 6000 bps, 2-PSK
 - 6000 bps, 4-PSK
 - 6000 bps, 8-PSK
 - 4000 bps, 4-QAM
 - 6000 bps, 16-QAM
 - 36.000 bps, 64-QAM
54. Determine a taxa de modulação para as taxas de transmissão e combinações de bits:
55. Determine a taxa de transmissão para cada taxa de modulação e tipo de modulação:
- 1000 baud, FSK
 - 1000 baud, ASK
 - 1000 baud, 8-PSK
 - 1000 baud, 16-QAM
56. Desenhe a constelação para os seguintes esquemas:
- ASK, amplitudes 1 e 3V

- b. 2-PSK, amplitude de 1V a 0° e 180°
57. Dados partem de uma fonte com valores distribuídos numa faixa entre $-1,0\text{V}$ e $+1,0\text{V}$. Para que valores binários os dados 0,91; -0,25; 0,56 e 0,71V são transformados se é utilizada uma regra de quantização de 8 bits?
58. Os pontos de dados de uma constelação estão em $(4,0)$ e $(6,0)$. Desenhe a constelação. Mostre a amplitude e a fase em cada ponto. Essa constelação representa ASK, PSK ou QAM? Quantos bits por modulação podem ser enviados nesta constelação?
59. Repita o Exercício 58 se os pontos estão em $(4,5)$ e $(8,10)$.
60. Repita o Exercício 58 se os pontos estão em $(4,0)$ e $(-4,0)$.
61. Repita o Exercício 58 se os pontos estão em $(4,4)$ e $(-4,4)$.
62. Repita o Exercício 58 se os pontos estão em $(4,0)$, $(4,4)$, $(-4,0)$ e $(-4,4)$.
63. A constelação na Figura 5.32 representa ASK, FSK, PSK ou QAM?
64. A constelação na Figura 5.33 representa ASK, FSK, PSK ou QAM?
65. A constelação na Figura 5.34 representa ASK, FSK, PSK ou QAM?
66. A constelação na Figura 5.35 representa ASK, FSK, PSK ou QAM?

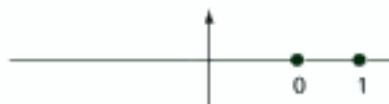


Figura 5.32 Exercício 63.

67. Uma constelação pode possuir 12 pontos? Justifique sua resposta.
68. Uma constelação pode possuir 18 pontos? Justifique sua resposta.
69. É possível definir uma regra geral para o número de pontos numa constelação?
70. Se o número de pontos numa constelação é 8, quantos bits podemos enviar por sinalização?
71. Determine a largura de banda necessária para cada uma das seguintes estações de AM. Desconsidere as regras do FCC.
- Sinal modulante com uma largura de banda de 4 kHz
 - Sinal modulante com uma largura de banda de 8 kHz
 - Sinal modulante com freqüências entre 2000 e 3000 Hz
72. Determine a largura de banda necessária para cada uma das seguintes estações de FM. Desconsidere as regras do FCC.
- Sinal modulante com uma largura de banda de 12 kHz
 - Sinal modulante com uma largura de banda de 8 kHz
 - Sinal modulante com freqüências entre 2000 e 3000 Hz.

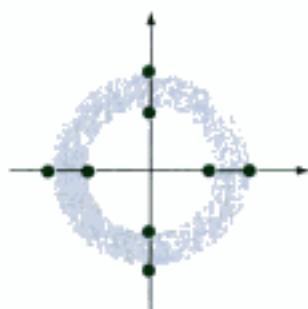


Figura 5.34 Exercício 65.

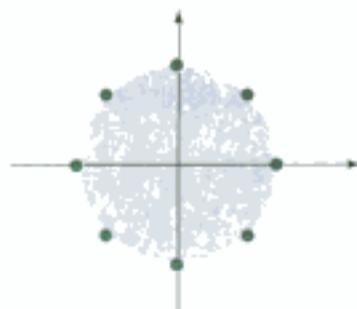


Figura 5.33 Exercício 64.

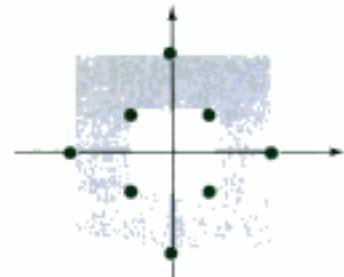


Figura 5.35 Exercício 66.

Multiplexação

Podemos compartilhar um meio que interliga dois dispositivos toda vez que a largura de banda desse meio for maior que a largura de banda necessária à comunicação entre os dispositivos. Denominamos **multiplexação** o conjunto de técnicas que permitem a transmissão simultânea de múltiplos sinais através de um único *link* de dados.

Em geral, quando o fluxo de dados e das telecomunicações cresce, aumenta-se o tráfego nas linhas de transmissão. Podemos suprir este aumento no tráfego adicionando continuamente linhas de transmissão individuais cada vez que um novo canal se fizer necessário, ou então, instalando *links* banda larga e utilizando-o para transportar múltiplos sinais. Como está descrito no Capítulo 7, hoje em dia a tecnologia permite utilizarmos meios de transmissão de largura de banda muito elevada, tais como a fibra óptica e os *links* de microondas terrestre ou via satélite. Cada um desses meios tem uma largura de banda excedente, muito além do que é realmente necessário para uma transmissão dos sinais típicos entre dois dispositivos. Se a largura de banda de um *link* for maior que a largura de banda necessária aos dispositivos conectados a ela, essa banda está sendo desperdiçada. Um sistema eficiente maximiza a utilização de todos os recursos; a largura de banda é um dos recursos mais preciosos de que dispomos na comunicação de dados.

As quatro linhas de entrada no lado esquerdo da figura direcionam o fluxo de transmissão para a saída do bloco denominado **multiplexador (MUX)**, onde são combinados em uma única cadeia de dados (MUX = muitas em uma). No lado receptor, a cadeia de dados incidente é introduzida num **demultiplexador (DEMUX)** que recupera de volta as componentes de dados agrupadas pelo MUX (DEMUX = uma em muitas) e as direciona para as linhas de saída correspondentes.

Na Figura 6.1, a palavra *link* refere-se ao caminho físico. A palavra **canal** refere-se à porção de um *link* que suporta uma transmissão entre um certo par de linhas. Um *link* pode ser composto de n canais.

Os sinais podem ser multiplexados através de uma das seguintes técnicas: Multiplexação por Divisão de Freqüência (Frequency-Division Multiplexing – FDM), Multiplexação por Divisão de Comprimento de Onda (Wave-Division Multiplexing – WDM) e Multiplexação por Divisão do Tempo (Time-Division Multiplexing – TDM). As duas primeiras são técnicas utilizadas para transmitir sinais analógicos e a terceira para transmitir sinais digitais (veja Figura 6.2).



Figura 6.1 Divisão de um *link* em canais.

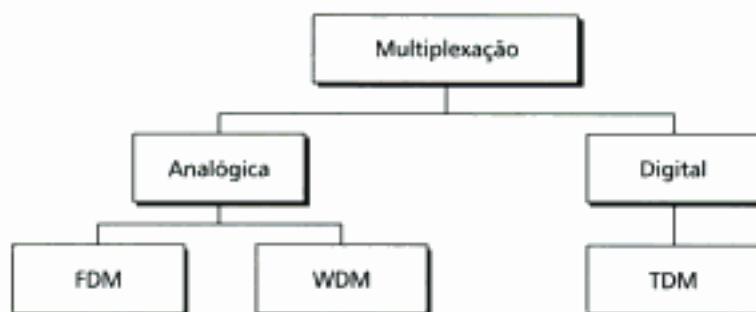


Figura 6.2 Tipos de multiplexação.

6.1 FDM

A **Multiplexação por Divisão de Freqüência (FDM)** é uma técnica analógica passível de utilização quando a **largura de banda** de um *link* (em hertz) é maior que as larguras de banda combinadas de todos os sinais a serem transmitidos. Na FDM, os sinais gerados em cada dispositivo transmissor são modulados em portadoras de freqüências diferentes. Em seguida, estes sinais são combinados em um único sinal composto para serem transportados através do *link*. As freqüências das portadoras são separadas através de bandas, suficientes para a perfeita acomodação do sinal modulado. Estas bandas são os canais através dos quais viajam vários sinais. Os canais devem ser separados por faixas de separação, inutilizadas propositadamente (**as bandas de proteção/segurança**), para prevenir a sobreposição dos sinais. Além disso, as freqüências das portadoras não devem interferir com as freqüências originais da transmissão de dados. Não seguir a condição anterior pode resultar no insucesso da recuperação dos sinais originais.

A Figura 6.3 ilustra a idéia conceitual da técnica FDM. Nesta ilustração, o caminho de transmissão é dividido entre três outros, cada qual representando um canal capaz de suportar uma transmissão. Façamos uma analogia: imagine um ponto onde três ruas individuais convirjam para formar uma única via expressa de três pistas. A cada uma das três ruas corresponde uma única pista na via expressa. Todo carro que entra na via expressa, partindo de uma das ruas, ainda possui uma pista própria e pode viajar sem interferir no fluxo de carros nas outras pistas.

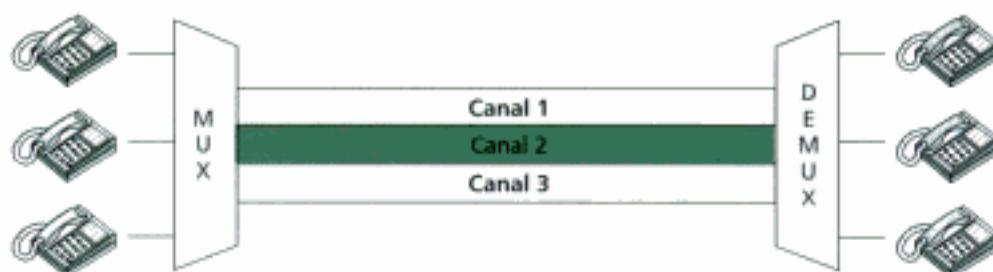


Figura 6.3 FDM.

FDM é uma técnica de multiplexação analógica que combina sinais.

Processo de Multiplexação

A Figura 6.4 ilustra conceitualmente um processo de multiplexação. A técnica FDM é um processo analógico, ilustrado através de linhas telefônicas suportando sinais de entrada oriundos de aparelhos telefônicos. Cada telefone gera um sinal analógico numa faixa de freqüência bastante similar. Dentro do MUX, estes sinais são modulados em portadoras de freqüências diferentes (f_1, f_2 e f_3). Então, os sinais modulados resultantes são combinados para formar um sinal composto único que é enviado através de um *link* que possui largura de banda suficiente para acomodá-lo.

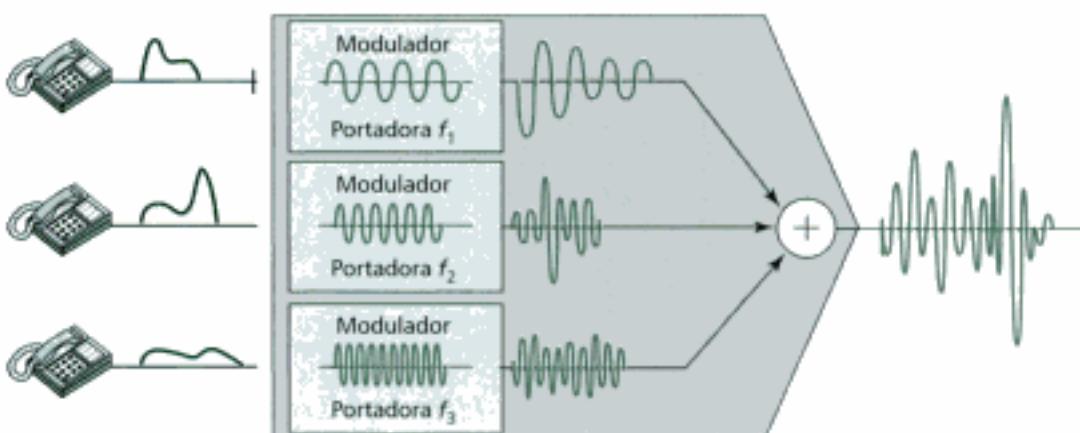


Figura 6.4 Processo FDM.

Processo de Demultiplexação

O processo de demultiplexação utiliza uma série de filtros para decompor o sinal multiplexado nas componentes constituintes. Os sinais individuais são então inseridos em um demodulador que os separa em portadoras e os repassa aos receptores na ponta de saída. A Figura 6.5 ilustra o processo de demultiplexação FDM, outra vez utilizando três linhas telefônicas como os dispositivos de comunicação.

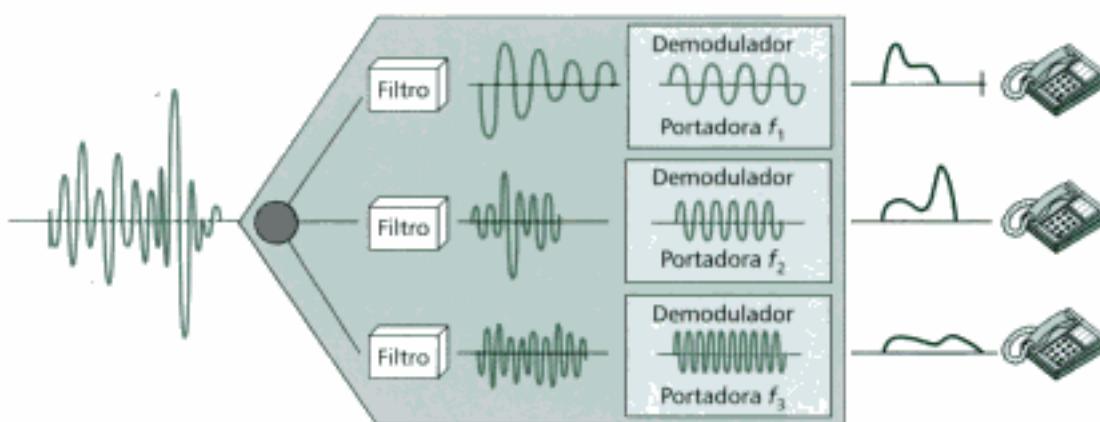


Figura 6.5 Exemplo de demultiplexação FDM.

Exemplo 1

Assuma que o canal de voz ocupa uma largura de banda de 4 kHz. Desejamos combinar três canais de voz em um link cuja largura de banda vale 12 kHz (20 a 32 kHz). Apresente os blocos com a configuração dos processos de multiplexação/demultiplexação, sem mostrar banda de proteção, no domínio da freqüência.

Solução

Desloque (module) cada um dos três canais de voz numa banda diferente, conforme Figura 6.6.

Reservamos a banda de 20 a 24 kHz para o primeiro canal, de 24 a 28 kHz foi reservado para o segundo canal e de 28 a 32 kHz acomoda o terceiro canal. Então, combinamos os três canais de acordo com a Figura 6.6. No receptor, cada canal recebe na entrada todo o sinal e, usando um filtro para separá-lo, seleciona apenas o sinal de interesse naquele canal. O primeiro canal usa um filtro que permite a passagem de freqüências entre 20 e 24 kHz e filtra todas as outras freqüências. O segundo canal utiliza um filtro que permite a passagem das freqüências entre 24 e 28 kHz e o terceiro canal utiliza um filtro que permite a passagem das freqüências entre 28 e 32 kHz. Em seguida, cada canal desloca a freqüência de volta para zero.

Exemplo 2

Cinco canais de 100 kHz de banda são multiplexados juntos. Qual é a menor largura de banda do link se necessitamos de uma banda de proteção de 10 kHz entre os canais para prevenir interferências?

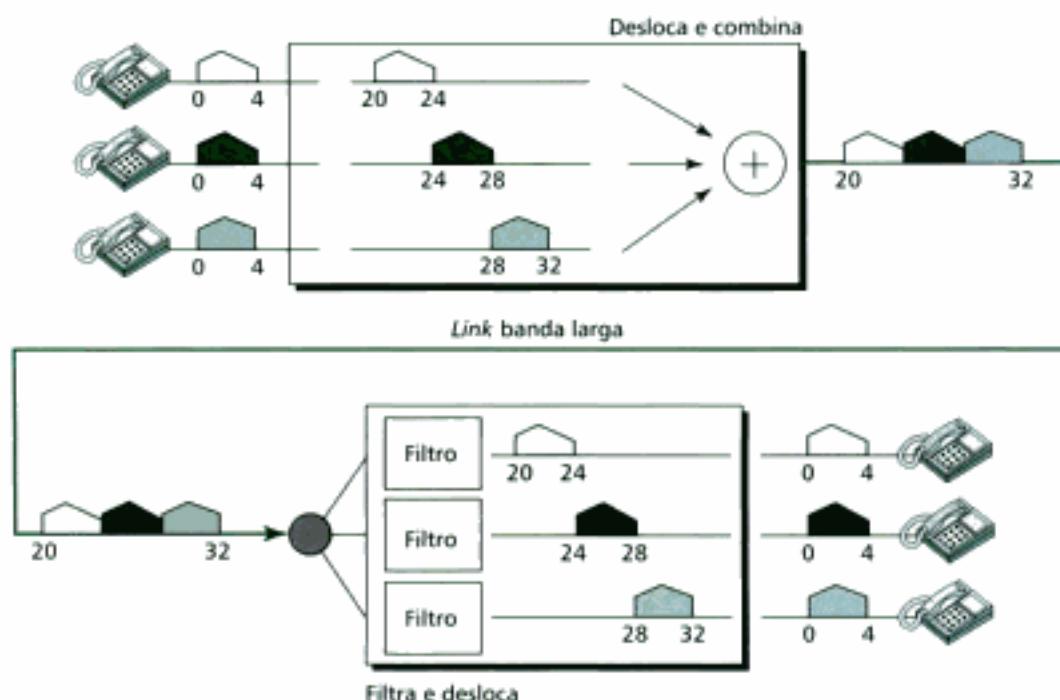


Figura 6.6 Exemplo 1.

Solução

Para cinco canais necessitamos de pelo menos quatro bandas de proteção. Isto significa que a largura de banda requerida no enunciado deve ser: $5 \times 100 + 4 \times 10 = 540 \text{ kHz}$ (como mostra a Figura 6.7.).

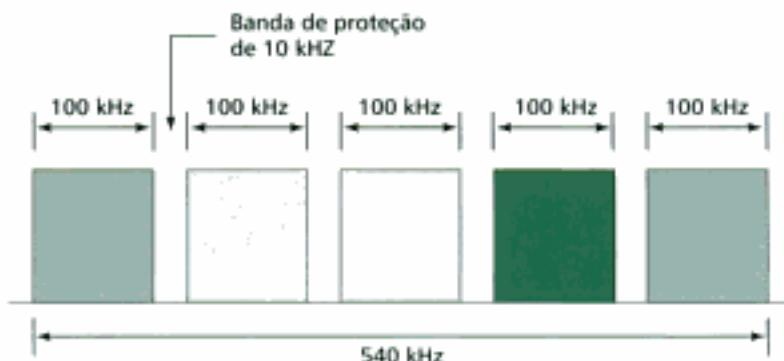


Figura 6.7 Exemplo 2.

Exemplo 3

Quatro canais de dados (digitais), cada um transmitindo 1 Mbps, usam um canal de satélite de 1 MHz. Propõa uma configuração adequada usando a técnica FDM.

Solução

O canal de satélite é analógico. Dividimos esse canal em quatro canais, cada qual com uma largura de banda de 250 kHz. Cada canal digital de 1 Mbps é modulado de tal forma que 4-bits são modulados a 1 MHz. Uma solução imediata é a modulação 16-QAM. A Figura 6.8 ilustra uma das configurações possíveis.

A Hierarquia Analógica

Para maximizar a eficiência da infra-estrutura das redes de telefonia pública, as companhias telefônicas utilizam tradicionalmente sinais multiplexados partindo das linhas de menor largura de banda para as linhas de maior largura de banda. Desse modo, muitas linhas comuta-

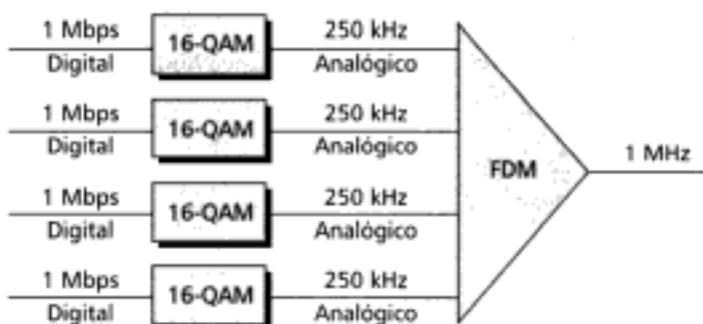


Figura 6.8 Exemplo 3.

das ou alugadas podem ser combinadas dentro de poucos, mas poderosos, canais. As linhas analógicas utilizam a FDM.

Um desses sistemas hierárquicos foi adotado pela AT&T que os denominou grupo, supergrupo, grupo mestre e grupo jumbo (veja Figura 6.9).

Nesta hierarquia analógica, o primeiro nível da hierarquia é formado de 12 canais de voz multiplexados em linhas banda larga criando um **grupo**. Um grupo possui uma largura de banda de 48 kHz e suporta 12 canais de voz.

No segundo nível da hierarquia, até 5 grupos podem ser multiplexados para criar um sinal composto originando um **supergrupo**. Um supergrupo possui uma largura de banda de 240 kHz e suporta até 60 canais de voz. Supergrupos podem ser montados através de 5 grupos ou por 60 canais de voz.

No próximo nível dessa hierarquia (o terceiro), 10 supergrupos são multiplexados para criar um **grupo mestre**. Um grupo mestre deve ter uma largura de banda de 2,40 MHz, devidamente separadas por banda de proteção que aumentam a largura de banda do canal para 2,52 MHz. Grupos mestres suportam até 600 canais de voz.

Finalmente, 6 grupos mestre podem ser combinados para formar o último nível da hierarquia, o **grupo jumbo**. Um grupo jumbo tem 15,12 MHz ($6 \times 2,52$ MHz), mas tipicamente são acrescidas bandas de proteção que o elevam para 16,984 MHz de modo a evitar interferência entre as bandas dos grupos mestres.

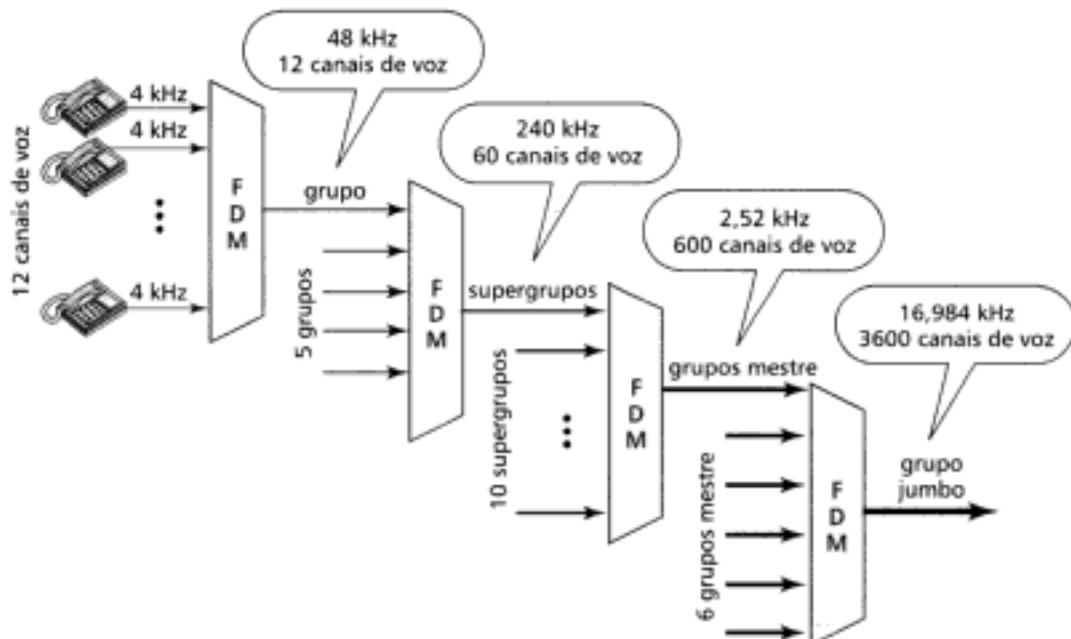


Figura 6.9 Hierarquia analógica.

Outras Aplicações FDM

Uma aplicação muito comum da FDM é a transmissão em multidifusão AM e FM. O rádio utiliza o ar como meio de transmissão. Uma banda especial (de 530 a 1700 kHz) é atribuída às estações AM. Todas as estações precisam compartilhar essa banda. De acordo com o Capítulo 5, cada estação AM necessita de uma largura de banda de 10 kHz para operar. Além disso, as estações utilizam portadoras em freqüências diferentes, onde na geração do sinal ocorre o deslocamento e multiplexação do sinal. O sinal que é enviado ao meio (o ar) é uma combinação de todos os sinais. Um receptor recebe todos os sinais de todas as estações simultaneamente, então filtra somente a estação desejada. Sem esse tipo de multiplexação somente uma estação AM poderia realizar multidifusão nesse link comum (o ar).

A situação é muito parecida para a transmissão em multidifusão FM. Entretanto, a transmissão FM utiliza uma banda mais larga (88 a 108 MHz) e cada estação necessita de uma largura de banda maior (200 kHz).

Outra aplicação interessante da FDM é na multidifusão dos canais de TV. Cada canal de TV possui uma largura de banda própria de 6 MHz.

A primeira geração de telefones celulares (ainda em operação em alguns países) também utiliza o conceito de FDM. A cada usuário é atribuído dois canais de 30 kHz, um para enviar e outro para receber sinal de voz. O sinal de voz, cuja largura de banda é 3 kHz (300 a 3300 kHz), é modulado usando técnicas de FM. Você deve se lembrar que um sinal FM deve possuir uma largura de banda de 10 vezes o valor da banda do sinal modulante, o que resulta numa largura de banda de 30 kHz (10×3 kHz) para cada canal. Desse modo, para cada usuário é cedida, pela estação base, uma largura de banda de 60 kHz, teoricamente disponível a todo instante para realização das chamadas.

Exemplo 4

A Advanced Mobile Phone System (AMPS) usa duas bandas. A primeira banda, localizada entre 824 e 849 MHz, é utilizada para enviar e a segunda, entre 869 e 894 MHz, é utilizada para receber. Os usuários possuem uma largura de banda de 30 kHz em cada direção. O sinal de voz de 3 kHz é modulado usando FM, gerando um sinal de 30 kHz. Quantas pessoas simultaneamente podem usufruir dos telefones celulares nesse sistema?

Solução

Cada banda possui 25 MHz. Se dividirmos 25 MHz em canais de 30 kHz, obtemos 833,33 canais. Na realidade, a banda é dividida em 832 canais. Desses canais, 42 são utilizados para controle, restando 790 canais para os usuários de telefones celulares. Discutiremos a AMPS em maiores detalhes no Capítulo 17.

Implementação

A FDM pode ser implementada muito facilmente. Em muitos casos, tal como nas transmissões de rádio e TV, não há nem a necessidade de multiplexadores ou demultiplexadores físicos. Nesses casos, como as estações seguem normas rígidas para transmitir sinais de multidifusão em freqüências diferentes através do meio (o ar), a multiplexação é implementada naturalmente durante a transmissão. Noutros casos, tal como nos sistemas de telefonia celular, estações base atribuem momentaneamente uma portadora de freqüência diferente aos usuários de telefone celular. É claro que não existe banda suficiente disponível, isto é, permanente para todos os usuários de telefones celulares ao mesmo tempo. Tão logo esses usuários terminam de utilizar a banda, ela torna-se disponível para outras chamadas.

6.2 WDM

A **Multiplexação por Divisão de Comprimento de Onda (Wave-Division Multiplexing – WDM)** foi desenvolvida para suportar transmissões de dados em velocidades altíssimas através de cabos de fibra óptica. A transmissão de dados dentro das fibras ocorre em velocidades muito superiores à transmissão em cabos metálicos. Entretanto, usar um cabo de fibra óptica para transmitir um único canal significa desperdiçar a largura de banda disponível. A multiplexação WDM permite-nos conectar várias linhas dentro de um único cabo ótico.

Conceitualmente, a WDM é muito parecida com a FDM, a não ser pelo fato de multiplexar e demultiplexar sinais óticos transmitidos através de canais de fibra ótica. A idéia é a mesma: combinamos vários sinais em diversas freqüências diferentes. Entretanto, a diferença mais significativa é que as freqüências envolvidas são muito altas.

A Figura 6.10 é uma visão conceitual de uma multiplexação/demultiplexação WDM. Sinais óticos de banda muito estreitas, proveniente de diversas fontes diferentes, são combinados de maneira a construir um canal "banda larga de luz". No lado do receptor, os canais voltam a ser separados pelo DEMUX.

WDM é uma técnica de multiplexação analógica capaz de combinar sinais óticos.

Embora a tecnologia envolvida seja muito complexa, a idéia é bastante simples. O mecanismo de funcionamento da WDM é simplesmente maravilhoso. No MUX, desejamos combinar múltiplas fontes luminosas em um único sinal de luz e no DEMUX separá-los novamente. O modo mais simples de convergir sinais luminosos e separá-los de volta é utilizar prismas. Da física básica, você deve lembrar que os prismas desviam feixes luminosos com base no ângulo de incidência e na freqüência do feixe. Através dessa técnica, um MUX é capaz de combinar muitos feixes luminosos incidentes, cada qual contendo uma estreita banda de freqüência, em um feixe de saída com um banda de freqüências muito larga. O DEMUX ótico reverte o processo separando os canais. A Figura 6.11 ilustra a idéia da WDM.

Uma aplicação da WDM é a rede SONET, onde muitas linhas de fibra ótica são multiplexadas e demultiplexadas. Discutiremos a SONET no Capítulo 9.

Uma nova técnica, denominada **DWDM (Dense WDM)**, é capaz de multiplexar um número grande de canais espaçados muito próximos uns dos outros. Essa técnica alcança índices de eficiência extraordinários.



Figura 6.10 WDM.

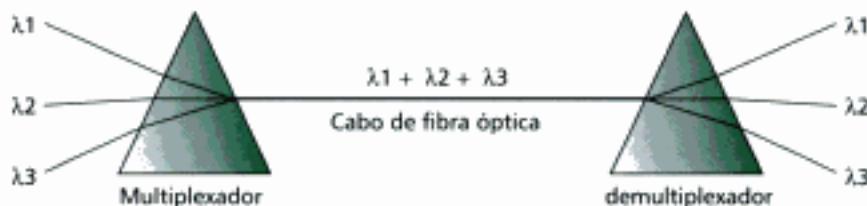


Figura 6.11 Prismas na multiplexação e demultiplexação WDM.

6.3 TDM

A **Multiplexação por Divisão de Tempo (Time-Division Multiplexing – TDM)** é um processo digital onde muitas conexões compartilham um link banda larga. Em vez de compartilhar uma porção da banda, como na FDM, toda a banda é entregue a um canal num determinado intervalo de tempo. Cada conexão ocupa o link durante uma porção do tempo. A Figura 6.12 é uma visão pictórica da TDM. Perceba que um mesmo link é utilizado, como na FDM. Porém, aqui o link é mostrado dividido no domínio do tempo, não no domínio da freqüência. Na figura, as porções dos sinais 1, 2, 3 e 4 ocupam seqüencialmente o link.

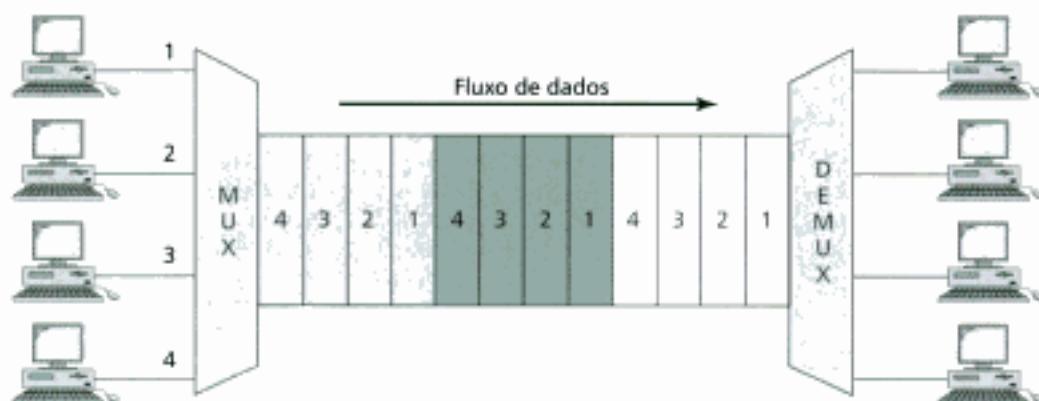


Figura 6.12 TDM.

TDM é uma técnica de multiplexação digital para combinar dados.

Time Slots e Frames

O fluxo de dados de cada conexão é dividido em unidades e o link combina uma unidade de cada conexão para montar um *frame* ou quadro. O tamanho de uma unidade de dados pode variar de 1 bit a milhares de bits. Para n conexões de entrada, um *frame* é organizado em pelo menos n *time slots* ou fatias de tempo, onde em cada *slot* é transportada uma unidade de dados de cada conexão. A Figura 6.13 mostra um exemplo com $n = 3$.

Na técnica TDM, a taxa de transmissão de dados no link que transporta dados das n conexões deve ser n vezes a taxa de transmissão de dados de uma conexão isolada para garantir o fluxo estável de dados. Consequentemente, o tempo de duração de uma unidade de dados numa conexão é n vezes o tempo de duração de um *time slot* num *frame*. Se considerarmos que o tempo de duração de um bit e a taxa de transmissão são recíprocos entre si, a exigência acima faz sentido. Na mesma Figura 6.13, a taxa de dados do link é 3 vezes a taxa de dados de uma conexão. Do mesmo modo, o tempo de duração de uma unidade numa conexão é 3 vezes o *time slot* (a duração de uma unidade num link). Na figura, representamos os dados antes da multiplexação com um tempo de 3 vezes o tempo após a multiplexação. Isto ilustra o fato de que cada unidade realmente deve durar 3 vezes mais antes da multiplexação do que depois.

Os *time slots* são agrupados em *frames*. Um *frame* consiste de um agrupamento completo de *time slots*, sendo que um *slot* é dedicado a cada dispositivo transmissor. Num sistema com n linhas de entrada, cada *frame* possui n *slots* e cada *slot* é destinado a transportar dados de uma linha de entrada específica.

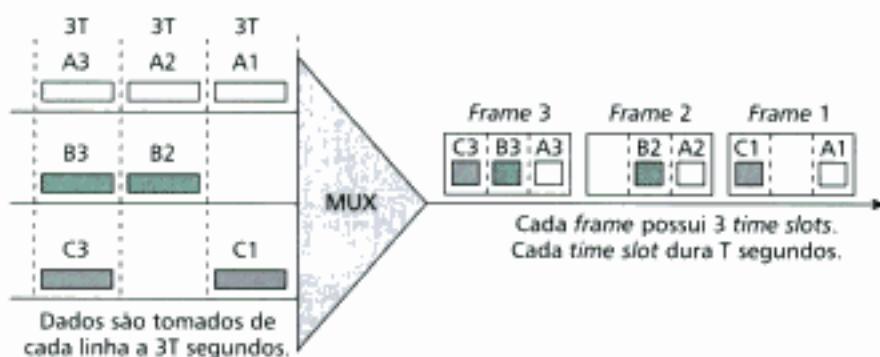


Figura 6.13 TDM.

Na multiplexação TDM, a taxa de transmissão de dados do link deve ser n vezes maior e o tempo de duração da unidade de dados n vezes menor.

Exemplo 5

Quatro conexões de 1 kbps são multiplexadas via TDM. Uma unidade de dados contém somente 1 bit. Determine (1) o tempo de duração de 1 bit antes da multiplexação (2) a taxa de transmissão do link (3) a duração de um time slot e (4) o tempo de duração de um frame.

Solução

Vamos responder os itens na ordem que eles foram solicitados:

1. O tempo de duração de 1 bit antes da multiplexação vale $1/1 \text{ kbps}$ ou $0,001 \text{ s}$ ou 1 ms .
2. A taxa de transmissão de dados do link deve ser 4 vezes a taxa de dados de uma conexão isolada, ou seja, 4 kbps.
3. A duração de cada time slot é um quarto do tempo de duração de cada bit antes da multiplexação, ou seja, $(1/4)\text{ms}$ ou $250 \mu\text{s}$. Note que esse mesmo resultado pode ser encontrado partindo da taxa de dados do link (4 kbps). O tempo de duração de um bit é o recíproco da taxa de dados do link, isto é, $(1/4)\text{kbytes/s}$ ou $250 \mu\text{s}$.
4. A duração de um frame é sempre a mesma duração de uma unidade de dados antes da multiplexação, nesse caso, 1 ms. Ainda, podemos determinar esse valor de outra maneira. Cada frame é constituído de quatro time slots. Então, o tempo de duração de um frame é $4 \times 250 \mu\text{s} = 1 \text{ ms}$.

Intercalando Sinais

A TDM pode ser visualizada através de duas chaves rotativas, uma no lado da multiplexação e outra no lado da demultiplexação. As duas chaves devem estar sincronizadas e girarem na mesma velocidade, mas em direções opostas. Quando a chave fecha numa conexão no lado do MUX, torna possível a transmissão de unidades de dados dessa conexão para o link. Este processo é denominado **intercalação de sinais**. Quando a chave fecha no lado do DEMUX, possibilita ao link repassar unidades de dados à conexão de saída que deve recebê-los.

A Figura 6.14 mostra o processo de chaveamento para a conexão ilustrada na Figura 6.13. Nesta figura, não assumimos nenhuma modificação na ordem do chaveamento para que os dados da primeira conexão do MUX saiam do DEMUX através da primeira conexão. As técnicas de chaveamento serão discutidas no Capítulo 8.

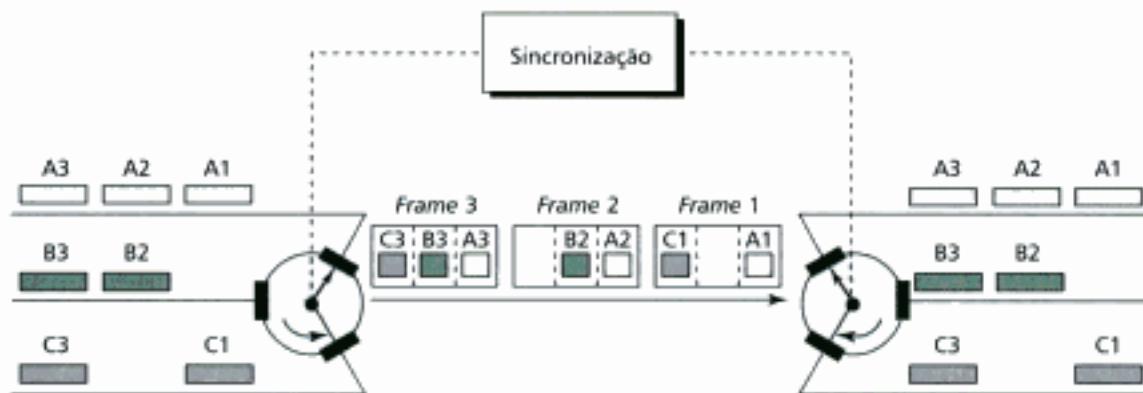


Figura 6.14 Intercalando slots.

Exemplo 6

Quatro canais são multiplexados usando TDM. Se a capacidade individual dos canais for 100 bytes/s e multiplexarmos 1 byte por canal, apresente (1) o esquema de viagem dos frames no link (2) o tamanho de cada frame (3) o tempo de duração de um frame e (4) a taxa de transmissão de dados do link.

Solução

O MUX é mostrado na Figura 6.15. Cada frame transporta 1 byte de cada canal. O tamanho de cada frame é 4 bytes ou 32 bits. Como cada canal é capaz de enviar 100 bytes/s e um frame transporta 1 byte de cada canal, a taxa de transmissão de frames no canal deve ser 100 frames/s. O tempo de duração de um frame vale $(1/100)\text{s} = 0,01 \text{ s} = 10 \text{ ms}$. O link transporta 100 frames/s e cada frame tem um tamanho de 32 bits. A taxa de transmissão de dados no link é 100×32 ou 3200 bps. De fato, isto equivale exatamente a 4 vezes a taxa de dados de cada canal, que é $100 \times 8 = 800 \text{ bps}$.

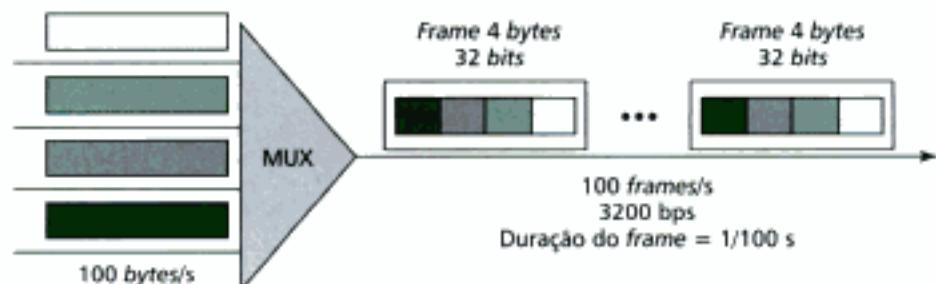


Figura 6.15 Exemplo 6.

Exemplo 7

Um MUX combina quatro canais de 100 kbps usando um *time slot* de 2 bits. Apresente a saída para quatro entradas arbitrárias. Qual é a taxa de transmissão de *frames*? Qual é o tempo de duração de um *frame*? Qual é a taxa de transmissão de dados? Qual é o tempo de duração de um bit?

Solução

A Figura 6.16 apresenta uma saída para quatro entradas arbitrárias. O link transporta 50.000 frames/s porque cada frame contém 8 bits por canal. O tempo de duração de um frame é $(1/50.000)$ s ou 20 µs. A taxa de transmissão de frames é 50.000 frames/s e cada frame transporta 8 bits. A taxa de transmissão de dados é $50.000 \times 8 = 400.000$ bps ou 400 kbps. Sendo assim, o tempo de duração de um bit é $(1/400.000)$ s ou 2,5 µs. Note que o tempo de duração de um frame é 8 vezes o tempo de duração de um bit porque cada frame está transportando 8 bits.

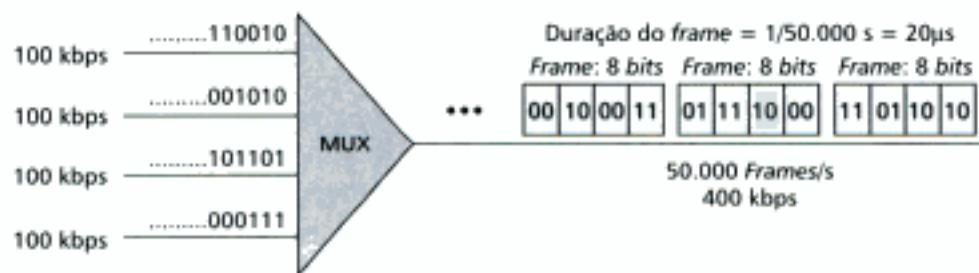


Figura 6.16 Exemplo 7.

Sincronização

Você deve ter notado que a implementação da TDM não é tão fácil quanto a FDM. O nível de sincronização entre o MUX e o DEMUX é maior nesse caso. Se os dois estiverem fora de sincronismo, um ou mais bits pertencentes a um canal podem ser recebidos pelo canal errado. Por esse motivo, um ou mais bits de sincronismo são adicionados usualmente no início de cada frame. Estes bits, denominados **framing bits**, seguem um padrão frame a frame que possibilita sincronizar o DEMUX com a cadeia de dados provenientes do MUX, de tal maneira que ele possa separar os *time slots* com precisão. Na maioria dos casos, esta informação adicional de sincronismo consiste de 1 bit por frame, alternando entre 0 e 1, como mostra a Figura 6.17.

Exemplo 8

Temos quatro fontes de dados, cada uma gerando 250 caracteres/s. Se chavearmos 1 unidade/s e adicionarmos 1 bit de sincronização em cada frame, determine (1) a taxa de dados da fonte, (2) o tempo de duração de cada caractere da fonte, (3) a taxa de transmissão de frames, (4) o tempo de duração de cada frame, (5) o número de bits em cada frame e (6) a taxa de transmissão de dados do link.

1. A taxa de transmissão de cada fonte é $250 \times 8 = 2000$ bps = 2 kbps.
2. Cada fonte envia 250 caracteres/s; consequentemente, a duração de um caractere é $(1/250)$ s ou 4 ms.



Figura 6.17 Bits de enchimento.

3. Cada *frame* possui um caractere de cada uma das fontes. Significa que o *link* deve ser capaz de enviar 250 *frames*/s para manter intacta a transmissão de dados de cada fonte.
4. A duração de cada *frame* é $(1/250)$ s ou 4 ms. Note que o tempo de duração de cada *frame* é o mesmo que a duração dos caracteres recebidos da cada fonte.
5. Cada *frame* transporta 4 caracteres e um bit extra de sincronização. Isto significa que o tamanho de cada *frame* corresponde a $4 \times 8 + 1 = 33$ bits.
6. O *link* transmite 250 *frames*/s e cada *frame* é composto de 33 bits. Isto significa que a taxa de transmissão de dados do *link* é 250×33 ou 8.250 bps. Perceba que a taxa de transmissão do *link* é maior que as taxas de transmissão combinadas dos quatro canais. Adicionando as taxas dos quatro canais chegamos a 8000 bps. Como 250 *frames* estão viajando por segundo no canal e cada um contém um bit extra de sincronização, precisamos adicionar 250 bps, introduzidos pelos bits extras, de modo a perfazer a taxa de transmissão de dados do *link*.

Bits de Enchimento

Podemos multiplexar dados entre dispositivos de diferentes velocidades de transmissão. Por exemplo, um dispositivo A poderia utilizar um *time slot*, enquanto que um dispositivo mais rápido B poderia utilizar dois *time slots*. A quantidade de *slots* num *frame* e o número de linhas de entrada para os quais eles são atribuídos, permanecem constantes do começo ao fim para um certo sistema de multiplexação, mas os dispositivos a diferentes taxas de dados podem controlar quantidades diferentes de *slots*. Lembre-se, o tamanho do *time slot* é fixo. Por essa razão, para que esta técnica funcione, as diferentes taxas de transmissão devem ser múltiplas inteiras entre si. Por exemplo, podemos ajustar um dispositivo que é 5 vezes mais rápido que os outros dispositivos dando-lhe 5 *slots*, um para cada dispositivo mais lento. Pelas mesmas razões, não conseguimos ajustar um dispositivo que é 5,5 vezes mais rápido que os outros porque não podemos introduzir meio *time slot* em um *frame*.

Quando as velocidades não são múltiplas inteiras entre si, podemos ajustá-las para que elas se comportem assim, através de uma técnica denominada **bits de enchimento (padding)**. Nesta técnica, o MUX adiciona bits extras à cadeia de dados da fonte de modo a forçar relações inteiras entre as taxas de transmissão dos vários dispositivos. Exemplificando, se tivermos um dispositivo cuja a taxa de transmissão é 2,75 vezes a taxa de dados dos dispositivos mais lentos, podemos adicionar bits suficientes para que a taxa de dados aumente para 3 vezes, em relação aos outros dispositivos. Chegando ao MUX, os bits extra serão descartados.

Exemplo 9

Dois canais, o primeiro com uma taxa de transmissão de 100 kbps e o segundo com uma taxa de transmissão de 200 kbps, são multiplexados. Como isto pode ser feito? Qual é a taxa de transmissão de *frames*? Qual é o tempo de duração de um *frame*? Qual é a taxa de transmissão do *link*?

Solução

Podemos alocar um *slot* para o primeiro canal e dois *slots* para o segundo canal. Então, cada *frame* transportará 3 bits. A taxa de transmissão de *frames* será 100.000 *frames*/s, já que ele transporta 1 bit do primeiro canal. O tempo de duração de um *frame* é $(1/100.000)$ s ou 10 µs. A taxa de transmissão do *link* será $100.000 \text{ frames/s} \times 3 \text{ bits/frame}$ ou 300 kbps. Visto que cada *frame* transporta 1 bit do primeiro canal, a taxa de transmissão do primeiro canal é preservada. A taxa de transmissão do segundo canal também é preservada porque cada *frame* transporta 2 bits do segundo canal.

Níveis de Sinal Digital (Serviços DS)

As companhias telefônicas implementaram a TDM através de uma hierarquia de sinais digitais, denominada **Serviço de Sinal Digital (DS)**. A Figura 6.18 apresenta o fluxo de dados suportado em cada nível da hierarquia.

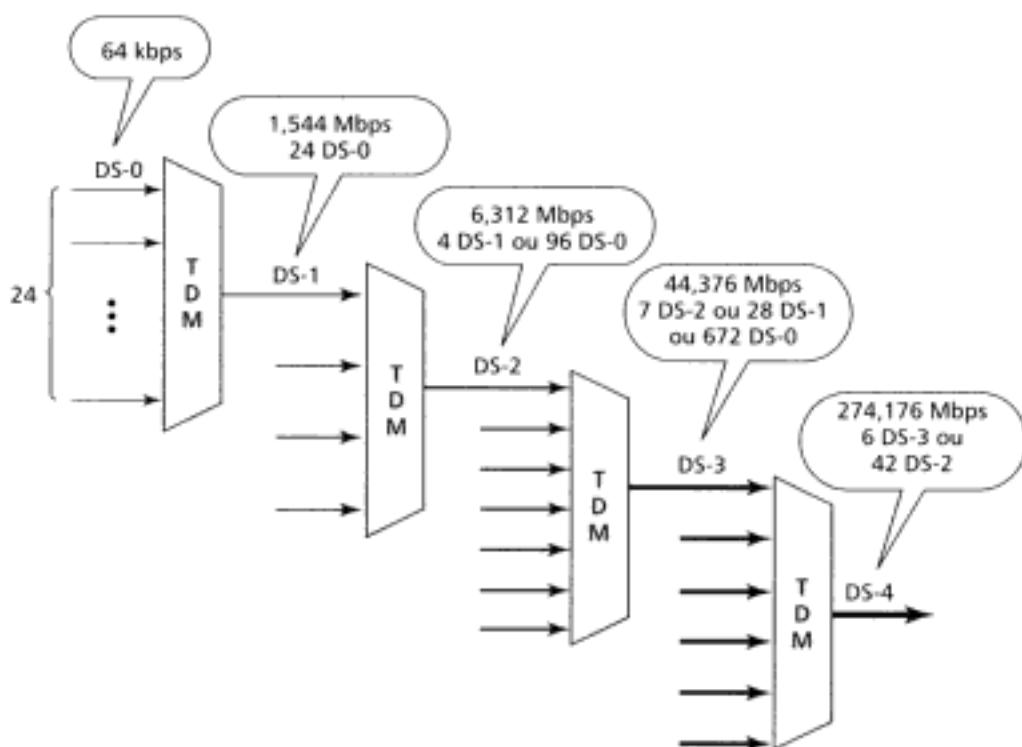


Figura 6.18 Hierarquia DS.

- Um serviço ou canal DS-0 é um canal digital único de 64 kbps.
- O sinal digital nível 1 (DS-1) é um serviço de 1,544 Mbps. Conforme mostra a Figura 6.18, são 24 canais DS-0. Assim, $24 \times 64 \text{ kbps} + 8 \text{ kbps de overhead}$ (sinalização) perfazem os 1,544 Mbps. Esse canal pode ser utilizado como um canal isolado de 1,544 Mbps, como um MUX de 24 canais DS-0 ou qualquer outra combinação desejada que perfeça no máximo os 1,544 Mbps.
- O sinal digital nível 2 (DS-2) é um serviço de 6,312 Mbps. De acordo com a figura, são 4 canais DS-1 ou 96 canais DS-0. Portanto, $96 \times 64 \text{ kbps} + 168 \text{ kbps de overhead}$ perfazem os 6,312 Mbps. Esse canal pode ser utilizado como um único serviço de transmissão a 6,312 Mbps, como 4 canais DS-1, 96 canais DS-0 ou qualquer combinação desses serviços.
- DS-3 é um serviço de 44,376 Mbps. São 7 canais DS-2 ou 28 canais DS-1 ou 672 canais DS-0. Logo, os 44,736 Mbps são $672 \times 64 \text{ kbps} + 1,368 \text{ Mbps de overhead}$.
- DS-4 é um serviço de 274,176 Mbps. São 6 canais DS-3 ou 42 canais DS-2 ou 168 canais DS-1 ou 4032 canais DS-0. Portanto, os 274,176 Mbps são $4032 \times 64 \text{ kbps} + 16,128 \text{ Mbps de overhead}$.

Hierarquia Digital T

Os canais DS-0 a DS-4 são os nomes dos serviços. Para implementá-los, as companhias telefônicas utilizam os **esquemas** ou **linhas T** (T1 – T4). Estas linhas possuem capacidades que perfazem exatamente as taxas de dados dos serviços DS-1 a DS-4 (veja Tabela 6.1).

TABELA 6.1 Taxa de transmissão das linhas DS e T

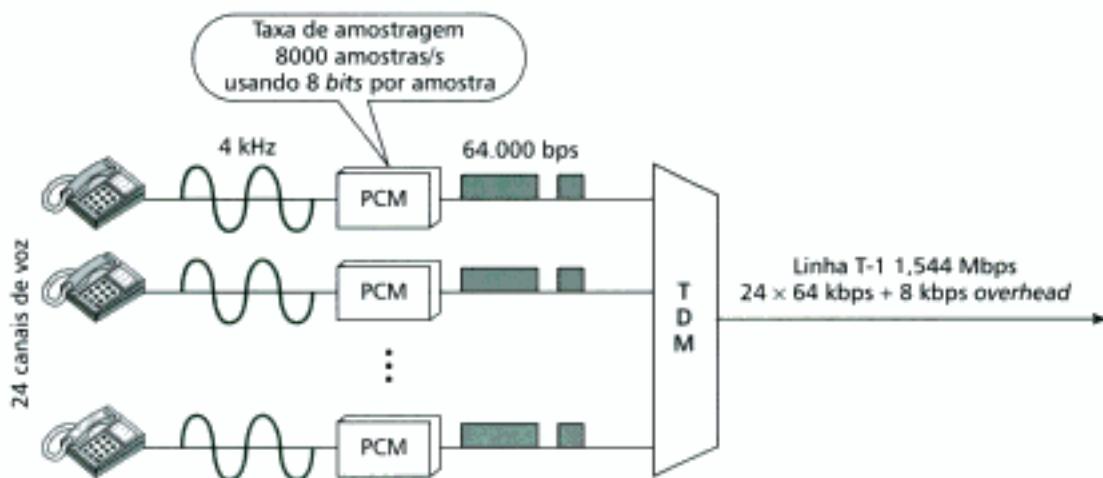
Serviço	Linha	Taxa (Mbps)	Canais de voz
DS-1	T-1	1,544	24
DS-2	T-2	6,312	96
DS-3	T-3	44,736	672
DS-4	T-4	274,176	4032

A linha T-1 é utilizada para implementar o serviço DS-1; a T-2 implementa o serviço DS-2 e assim por diante. Como mostra a Tabela 6.1, DS-0 não é oferecido como um serviço, mas ele tem sido definido como referência para a hierarquia.

Linhas T para Transmissão Analógica

As linhas T são totalmente digitais e foram projetadas para transmissão de dados digitais, áudio ou vídeo. Entretanto, elas também podem ser usadas para realizar transmissões analógicas (telefonia fixa), desde que os sinais analógicos sejam amostrados antes da TDM.

A possibilidade de utilizar as linhas T para transmitir sinais analógicos abriu um novo leque de serviços para as companhias telefônicas. No passado, quando uma organização desejava 24 linhas telefônicas separadas era necessário instalar 24 pares de fios da empresa até a central mais próxima. (Você se lembra daqueles filmes antigos mostrando um executivo ocupado com 10 linhas telefônicas sobre a mesa? Ou, das salas onde ficavam as centrais telefônicas antigas e por onde entrava um cabo grosso? Aqueles cabos continham um conjunto enorme de linhas (fios) separados.) Hoje, essa mesma organização pode combinar as 24 linhas numa única linha T-1. A Figura 6.19 ilustra como 24 canais de voz podem ser multiplexados numa linha T-1 (se necessário, estude novamente a codificação PCM no Capítulo 5).

**Figura 6.19** Linha T1 multiplexando linhas telefônicas.

O Esquema de Portadora T1

De acordo com os parágrafos anteriores, o serviço DS-1 requer um *overhead* de 8 kbps. Vamos examinar o formato de um *frame* de um canal T-1 (24 canais de voz) para entender como o *overhead* é calculado.

O *frame* utilizado numa linha T-1 tem 193 bits divididos em 24 *slots* de 8 bits cada, mais 1 bit extra de sincronização ($24 \times 8 + 1 = 193$); veja a Figura 6.20. Noutras palavras, cada *slot* contém um segmento de sinal de cada canal e 24 segmentos são intercalados num *frame*. Se uma linha T-1 transportar 8000 *frames*, a taxa de transmissão de dados é 1,544 Mbps ($193 \times 8000 = 1,544$ Mbps), ou seja, a capacidade da linha T-1.

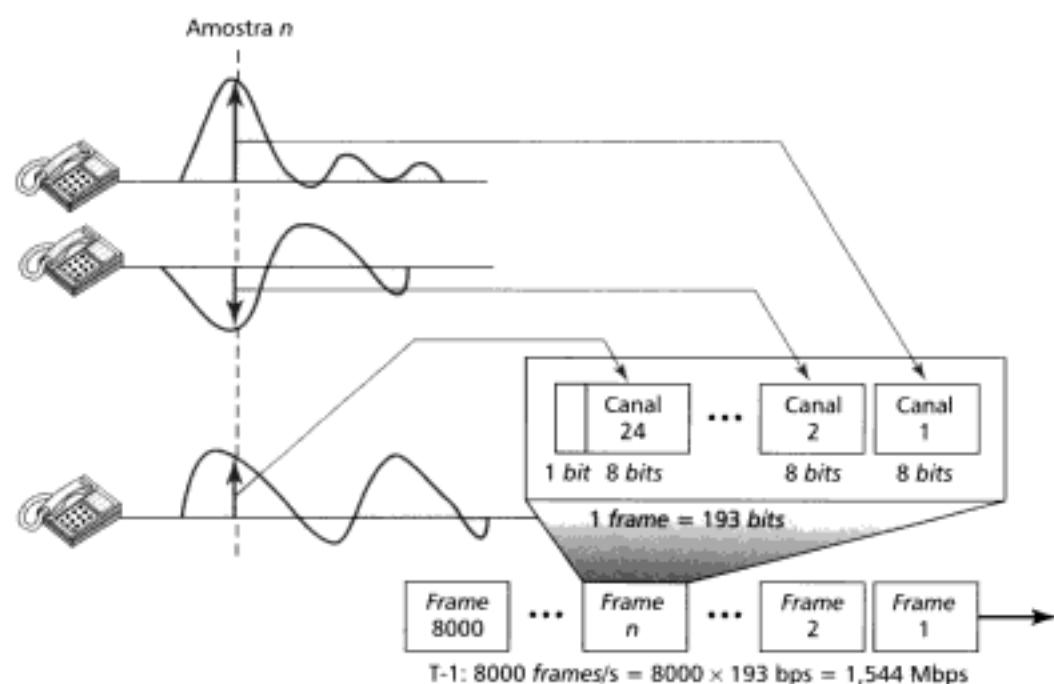


Figura 6.20 Estrutura do frame T-1.

Hierarquia Digital E

Os europeus* usam uma versão das linhas T denominada **linhas E**. Os dois sistemas são idênticos do ponto de vista conceitual, mas diferem em termos das capacidades. A Tabela 6.2 mostra as linhas E e as respectivas capacidades.

TABELA 6.2 Linhas E

Linhas E	Taxa (Mbps)	Canais de voz
E-1	2,048	30
E-2	8,448	120
E-3	34,368	480
E-4	139,264	1920

TDM Inverso

Como sugere o nome, a **multiplexação inversa** funciona de maneira oposta à multiplexação tradicional. Ela recebe uma cadeia de dados de alta velocidade na entrada e a quebra em várias linhas para poderem ser transmitidas, simultaneamente, em linhas de velocidades mais baixas, sem perdas na taxa de transmissão coletiva de dados (veja Figura 6.21).

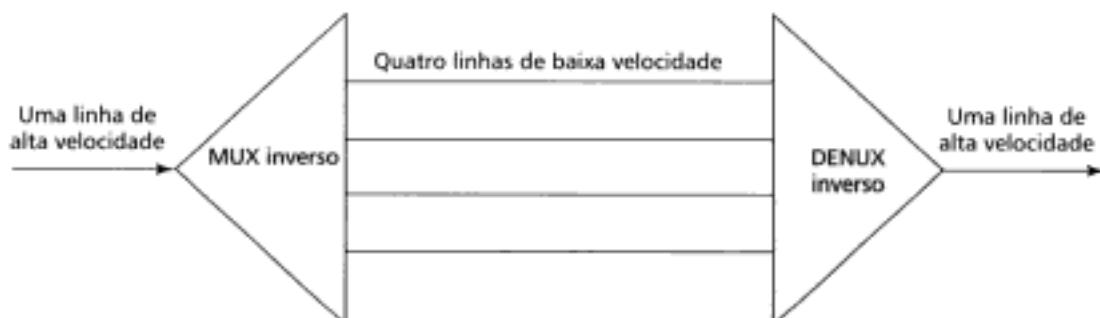


Figura 6.21 Multiplexação e multiplexação inversa.

* N. de R. T.: O Brasil segue o padrão europeu. A hierarquia digital das linhas T é um padrão americano.

Por que precisaríamos da multiplexação inversa? Imagine uma empresa que deseja transmitir dados, áudio e vídeo, e cada um desses objetos requerendo taxas de transmissão diferentes. Para transmitir áudio ela pode desejar um *link* de 64 kbps. Para enviar dados, ela pode necessitar de um *link* de 128 kbps e, para vídeo, pode precisar de um *link* de 1,544 Mbps. De modo a dispor de todos os serviços, a empresa tem duas opções. Alugar um canal de 1,544 Mbps de uma companhia telefônica e usar a capacidade completa da linha algumas vezes, o que não é algo eficiente do ponto de vista financeiro. Alugar vários canais separados de taxas de transmissão mais baixas. Contratando **banda sob demanda**, a empresa pode usar qualquer um desses canais quando e da forma que desejar. Transmissões de voz passam intactas em qualquer canal. Já as transmissões dos sinais de dados ou de vídeo podem ser quebradas e enviadas em duas ou mais linhas. Noutras palavras, os sinais de dados e vídeo podem ser multiplexados inversamente em muitas linhas.

Outras Aplicações da TDM

Algumas empresas de telefonia celular de segunda geração usam a TDM. Por exemplo, a versão digital da telefonia celular que citamos anteriormente divide a largura de banda disponível em bandas de 30 kHz e usa a FDM para combinar tais bandas. Em cada banda, pode ser aplicada a TDM de maneira tal que seis usuários, por exemplo, possam compartilhar a mesma banda. Significa que a banda de 30 kHz é dividida em seis *time slots* e os sinais de voz digitalizados dos usuários são inseridos nesses *slots*. Através da técnica TDM, o número de usuários em cada área pode ser aumentado significativamente. Trataremos a telefonia celular de segunda geração no Capítulo 17.

6.4 TERMOS-CHAVE

Banda de proteção	Hierarquia analógica
Banda sob demanda	Intercalação de sinais
<i>Bits</i> de enchimento (<i>padding</i>)	Linhas E
Canal	Linhas T
Demultiplexador (DEMUX)	<i>Link</i>
Dense Wave-Division Multiplexing (DWDM)	Multiplexação inversa
<i>Framing bit</i>	Multiplexador (MUX)
Frequency-Division Multiplexing (FDM)	Níveis de sinal digital
Grupo	Supergrupo
Grupo jumbo	Time-Division Multiplexing (TDM)
Grupo mestre	Wave-Division Multiplexing (WDM)

6.5 RESUMO

- Multiplexação é uma técnica de transmissão simultânea de muitos sinais através de um único *link* de dados.
- A Multiplexação por Divisão de Freqüência e a Multiplexação por Divisão de Comprimento de Onda (WDM) são técnicas de multiplexação de sinais analógicos, enquanto a *Time-Division Multiplexing* (TDM) é uma técnica de multiplexação digital.
- Na FDM, cada sinal é modulado numa portadora de freqüência diferente. Todas as portadoras juntas são combinadas para formar um novo sinal que é então enviado através do *link*.
- Na FDM, os multiplexadores modulam e combinam sinais enquanto os demultiplexadores decomprimem e demodulam sinais.
- Na TDM, sinais digitais de n dispositivos são intercalados uns com os outros para formar um *frame* de dados (*bits*, *bytes* ou outra unidade de dados qualquer).
- Na WDM, os multiplexadores combinam os vários canais de luz e os demultiplexadores descombinam os canais de luz.
- Na FDM, os canais de voz são divididos em bandas de 30 kHz e combinados. Na TDM, os canais de voz são divididos em *time slots* e combinados.
- Na FDM, os canais de voz são divididos em bandas de 30 kHz e combinados. Na TDM, os canais de voz são divididos em *time slots* e combinados.
- Empresas de telefonia usam a FDM para combinar canais de voz em grupos sucessivamente maiores de modo a aumentar a eficiência da transmissão.
- A WDM é similar conceitualmente à FDM. Entretanto, os sinais que são multiplexados na WDM são sinais de luz.
- Na TDM, sinais digitais de n dispositivos são intercalados uns com os outros para formar um *frame* de dados (*bits*, *bytes* ou outra unidade de dados qualquer).

- Bits de framing permitem que um MUX TDM sincronize dispositivos rápidos com dispositivos lentos.
- Sinal Digital (DS) forma uma hierarquia de sinais TDM.
- As linhas T (T-1 a T-4) são a implementação física dos serviços DS. Uma linha T-1 é constituída de 24 canais de voz.
- As linhas T são um padrão americano. O padrão europeu é denominado linhas E.
- Multiplexação inversa divide uma cadeia de dados de alta velocidade em muitas linhas de velocidades menores.

6.6 PRATIQUE OS CONHECIMENTOS ADQUIRIDOS

Questões de Revisão

1. Quais são as três maiores técnicas de multiplexação?
2. Como a FDM combina múltiplos sinais em um único?
3. Qual é o propósito da banda de proteção ou de segurança?
4. Como um sinal FDM é separado nas componentes originais?
5. Descreva o arranjo de agrupamento da hierarquia analógica.
6. Como a WDM se assemelha à FDM? De que maneira elas são diferentes?
7. Como a TDM faz para combinar muitos sinais em um único sinal?
8. Como um sinal TDM é separado nas componentes originais?
9. Comente sobre o tempo de duração de uma unidade de dados antes e depois do processo de TDM.
10. Descreva a hierarquia DS.
11. Como as linhas T se relacionam com os serviços DS?
12. Como as linhas T podem ser utilizadas para transmitir informação analógica?
13. Qual é a relação entre o número de slots num frame e o número de linhas de entrada para a TDM?
14. Os bits de enchimento é uma técnica FDM ou TDM? Os bits de enchimento são usados na FDM ou TDM?
15. O que é multiplexação inversa?

Questões de Múltipla Escolha

16. O compartilhamento de um meio/link por dois ou mais dispositivos é denominado _____.
 - a. Modulação
 - b. Codificação
 - c. Disciplina de linha
 - d. Multiplexação
17. Que técnica de multiplexação é capaz de transmitir sinais analógicos?
 - a. FDM
 - b. TDM
 - c. WDM
 - d. (a) e (c)
18. Que técnica de multiplexação é capaz de transmitir sinais digitais?
 - a. FDM
 - b. TDM
 - c. WDM
 - d. Nenhuma das anteriores
19. Que técnica de multiplexação desloca cada sinal para uma frequência diferente?
 - a. FDM
 - b. TDM
20. Na TDM, para n fontes de dados com a mesma taxa de dados, cada frame contém _____ slots.
 - a. n
 - b. $n + 1$
 - c. $n - 1$
 - d. 0 a n
21. Na TDM, a taxa de transmissão do link é usualmente _____ a soma das taxas de transmissão das fontes de dados.
 - a. Maior que
 - b. Menor que
 - c. Igual
 - d. 1 menor que
22. Na hierarquia FDM da AT&T, a largura de banda de cada tipo de grupo pode ser determinada multiplicando _____ e adicionando a banda extra de proteção.
 - a. O número de canais de voz por 4000 Hz.
 - b. A taxa de amostragem por 4000 Hz.

- c. O número de canais de voz por 8 bits/amostra.
 - d. A taxa de amostragem por 8 bits/amostra.
23. DS-1 a DS-4 são _____ enquanto T-1 a T-4 são _____.
 a. Serviços; multiplexadores
 b. Serviços; sinais
 c. Serviços; linhas
 d. Multiplexadores; sinais
24. Na linha T-1, ocorre a intercalação de _____.
 a. Bits
 b. Bytes
 c. DS-0
- d. Chaves
25. Bandas de proteção aumentam a largura de banda na técnica _____.
 a. FDM
 b. TDM
 c. (a) e (b)
 d. Nenhuma das anteriores
26. Que técnica de multiplexação manipula os sinais compostos de feixes luminosos?
 a. FDM
 b. TDM
 c. WDM
 d. Nenhuma das anteriores

Exercícios

27. Determine a largura de banda mínima do link com base nas informações fornecidas.
 a. Multiplexação FDM.
 b. Cinco linhas, cada qual requer 4000 Hz.
 c. Banda de proteção de 200 Hz separando as bandas.
28. Determine a largura de banda máxima do link com base nas informações fornecidas.
 a. Multiplexação FDM.
 b. Largura de banda total disponível = 7900 Hz.
 c. Três fontes de sinal.
 d. Banda de proteção de 200 Hz entre cada fonte de sinal.
29. Cinco fontes de dados são multiplexadas através da TDM. Cada fonte produz 100 caracteres/s. Assuma que a intercalação dos canais acontece em bytes e que cada frame requer um bit de sincronização, determine (1) a taxa de transmissão de frames, (2) a taxa de transmissão do link.
30. Dada a seguinte informação, desenhe frames TDM mostrando os caracteres de dados:
 a. Quatro fontes de sinal
 b. Fonte 1, mensagem: T E G
 c. Fonte 2, mensagem: A
 d. Fonte 3, mensagem:
 e. Fonte 4, mensagem: E F I L
31. Qual é o tempo de duração de um frame na linha T-1?
32. A linha T-2 oferece serviço a 6,312 Mbps. Por que este número não é $4 \times 1,544$ Mbps?
33. Na Figura 6.19, a taxa de amostragem é 8000 amostras/s. Por quê?
34. Se uma fibra óptica monomodo pode transmitir dados a 2 Gbps, quantos canais de telefone podem coexistir num mesmo cabo ótico?
35. Calcule o overhead (em bits) por canal de voz em cada linha T. Qual é a percentagem de overhead por canal de voz?
36. Três canais de comunicação, cada um usando 4 kHz, são multiplexados em freqüência, utilizando AM, de modo a anular a banda modulada inferior. Desenhe uma representação do sinal resultante no domínio da freqüência se as portadoras têm freqüência de 4, 10 e 16 kHz, respectivamente. Qual é a largura de banda do sinal resultante?
37. Apresente a representação no domínio da freqüência dos sinais resultantes em cada estágio da Figura 6.22. Assuma que não existem bandas de proteção. Escolha adequadamente as freqüências portadoras.
38. Multiplexamos 100 computadores usando TDM síncrona. Se cada computador transmite dados a 14,4 kbps, qual é a taxa de transmissão mínima da linha? Uma linha T-1 pode ser útil nesta situação?
39. Qual é a taxa de transmissão mínima de cada linha na Figura 6.23 se usarmos multiplexação TDM síncrona? Ignore os framing bits de sincronismo.

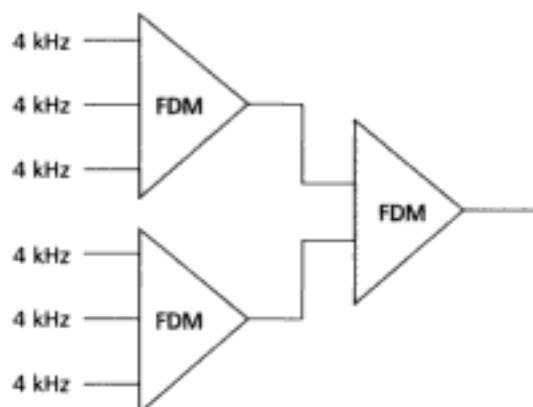


Figura 6.22 Exercício 37.

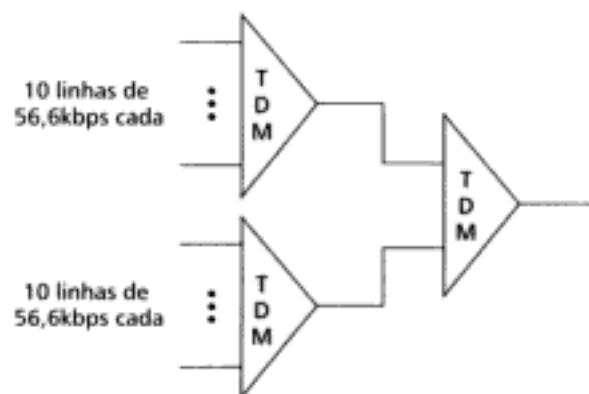


Figura 6.23 Exercício 39.

40. A Figura 6.24 mostra um MUX. Se o slot tem apenas 10 bits (3 bits tomados em cada canal mais 1 framing bit extra), qual é a configuração de bits da cadeia de saída? Qual é a taxa de transmissão de saída? Qual é o tempo de duração de cada bit na linha de saída? Quantos slots são enviados por segundo? Qual é o tempo de duração de cada slot?
41. A Figura 6.25 mostra um DEMUX. Se o slot de entrada tem 12 bits (ignore os framing bits), qual é a configuração de bits da cadeia de saída? Qual é a taxa de transmissão de cada linha de saída?
42. A Figura 6.26 mostra um MUX inverso. Se a taxa de taxa de transmissão de entrada for 15 Mbps, qual é a taxa de dados em cada linha? Podemos usar o serviço dos canais T-1 para este propósito? Ignore os framing bits.
43. Qual é o overhead gerado (número de bits extras por segundo) numa linha T-1?
44. Se desejamos conectar duas LANs Ethernet idênticas com taxas de dados de 10 Mbps, quantos canais T-1 são necessários para realizar essa conexão? Necessitaremos de multiplexação tradicional ou inversa? Apresente a configuração.

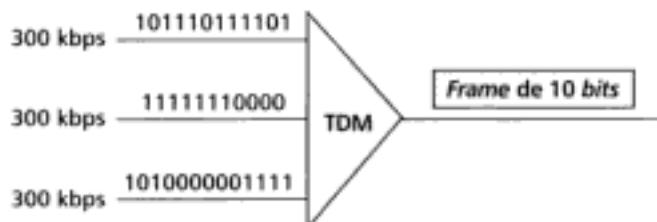


Figura 6.24 Exercício 40.

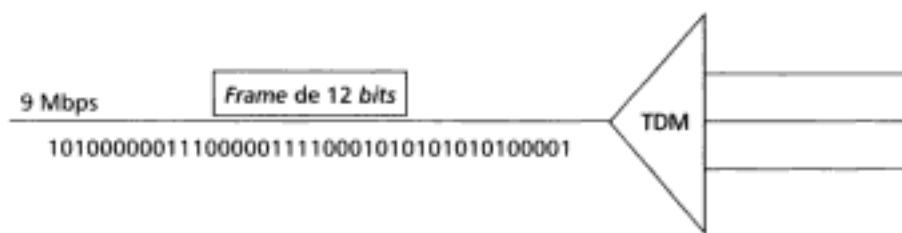


Figura 6.25 Exercício 41.

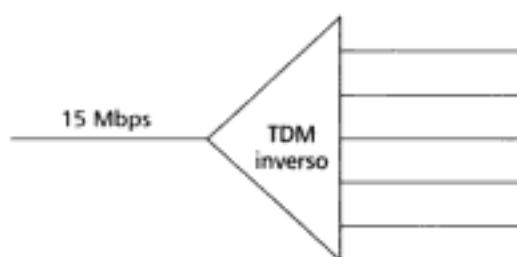


Figura 6.26 Exercício 42.

Meios de Transmissão

Examinamos muitas questões relacionadas à camada física nos Capítulos 3 a 6. Neste capítulo, estudaremos os meios de transmissão em si. Em qualquer modelo de camadas, os meios de transmissão situam-se logo abaixo da camada física, sendo controlado diretamente por essa camada. Podemos dizer, nos referindo aos modelos OSI e TCP/IP, que os meios de transmissão pertencem à "camada zero". A Figura 7.1 situa os meios de transmissão em relação à camada física.

De acordo com a discussão do Capítulo 3, computadores e dispositivos de telecomunicação genéricos usam sinais para representar dados. Estes sinais são transmitidos de um dispositivo para outro na forma de pacotes de energia eletromagnética, propagando-se através do meio de transmissão.

Energia eletromagnética é uma combinação vetorial dos campos elétrico e magnético, variáveis no tempo e mantendo ortogonalidade entre si, transportada por ondas de rádio, luz visível ou invisível (infravermelho ou ultravioleta), raios X, gama e cósmico de alta energia. Cada um desses elementos constitui uma faixa no **espectro eletromagnético**. Contudo, nem todas as faixas do espectro são utilizadas nas telecomunicações. Os meios definem que faixas do espectro são úteis para a transmissão. Entretanto, não existem tantos tipos diferentes de meios de transmissão. Por isso, nossas escolhas do tipo de meio em geral são limitadas.

Para os nobres propósitos das telecomunicações, as transmissões podem ser divididas em duas grandes categorias: guiadas e sem fio. Uma transmissão guiada utiliza um guia de onda como suporte dos sinais eletromagnéticos. Dentre os meios guias de onda podemos citar os cabos par trançado, coaxial e fibra óptica. Numa transmissão sem fio, geralmente utilizamos o ar como suporte dos sinais eletromagnéticos. A Figura 7.2 mostra o relacionamento entre os meios de transmissão.

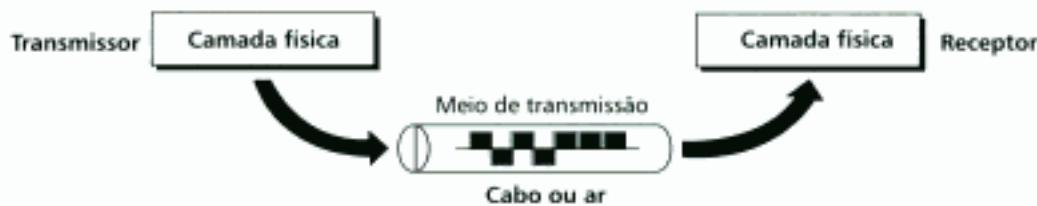


Figura 7.1 Meios de transmissão e a camada física.



Figura 7.2 Tipos de meios de transmissão.

7.1 TRANSMISSÃO GUIADA

Um guia de onda é um suporte físico especialmente projetado para acomodar e transportar sinais eletromagnéticos entre dois dispositivos. Na comunicação de dados, os meios guias de onda mais utilizados são os cabos **par trançado**, **coaxial** e **fibra óptica**. Um sinal viajando ao longo de qualquer um desses meios é direcionado e acomodado dentro dos limites físicos do meio. Tanto o par trançado quanto o cabo coaxial usam condutores metálicos (tipicamente de cobre) para transportar sinais na forma de corrente elétrica. Uma **fibra óptica** é um cabo de vidro que aceita e transporta sinais eletromagnéticos na forma de pulsos luminosos.

Cabo Par Trançado

Um par trançado consiste de um par de fios trançados, normalmente de cobre, cada qual envolvido por uma jaqueta de PVC isolante, como mostra a Figura 7.3.

Um dos fios do par funciona como suporte de transporte dos sinais entre transmissor e receptor, enquanto que o outro funciona como referência ou terra do sinal. O receptor baseia-se na diferença de potencial entre os dois níveis de tensão nos fios para interpretar a informação transmitida. Sem a referência não há como saber o que está se tentando transmitir num cabo metálico.

Além do mais, os sinais enviados do transmissor ao receptor podem estar sujeitos à interferência eletromagnética (ruído) e à diafonia (*crosstalk*), afetando ambos os fios, e criando um sinal indesejável. O receptor opera somente olhando a diferença entre os sinais indesejáveis. Isto significa que se o par de fios é afetado igualmente pelo ruído e *crosstalk*, o receptor ficará imune ao efeito global, porque a diferença no par de fios produzirá o cancelamento dos sinais indesejados.

Se o par de fios guarda um paralelismo dentro do cabo, o efeito dos sinais indesejáveis não é o mesmo em ambos fios, visto que eles estão em localizações diferentes relativamente às fontes de ruídos e de interferência eletromagnética. Isto resulta numa diferença não nula no receptor. Trançando os pares, as inversões periódicas entre os fios dos pares ocorre por igual dentro do cabo. A explicação do cancelamento do trançado é a seguinte: suponha que no par trançado um dos fios está mais próximo da fonte de ruído. Na próxima torcida do par ocorre o contrário, o fio que estava mais próximo da fonte de ruído trocou de posição com o par dele. O efeito trançado do par é aumentar a probabilidade de ambos os fios serem igualmente afetados pelas influências externas (ruído e *crosstalk*). Isto facilita a eliminação dos sinais indesejáveis no recep-



Figura 7.3 Cabo par trançado.

tor, pois ele olha sempre a diferença entre os dois fios. Partindo da discussão acima, está claro que o número de tranças por unidade de comprimento (p. ex., mm ou pol) determina a qualidade do cabo. Quanto mais tranças tiver um cabo, maior é a qualidade desse cabo.

UTP versus STP

O cabo de pares trançados mais utilizados nas comunicações é o **Par Trançado sem Blindagem (Unshielded Twisted-Pair – UTP)**. A IBM também produz uma versão do cabo de pares trançados para o uso em aplicações desenvolvidas por ela, denominado **Par Trançado Blindado (Shielded Twisted Pair – STP)**. O cabo STP tem uma blindagem eletromagnética metálica ou revestimento em malha de cobre em cada par de fios isolados do cabo. Embora a blindagem melhore a qualidade do cabo contra interferências provocadas por ruídos ou *crosstalk*, o cabo STP é tanto mais caro e tanto mais volumoso que o UTP. A Figura 7.4 ilustra a diferença entre UTP e STP. Nossa discussão focará inicialmente o cabo UTP porque STP raramente é utilizado fora do ambiente da IBM.

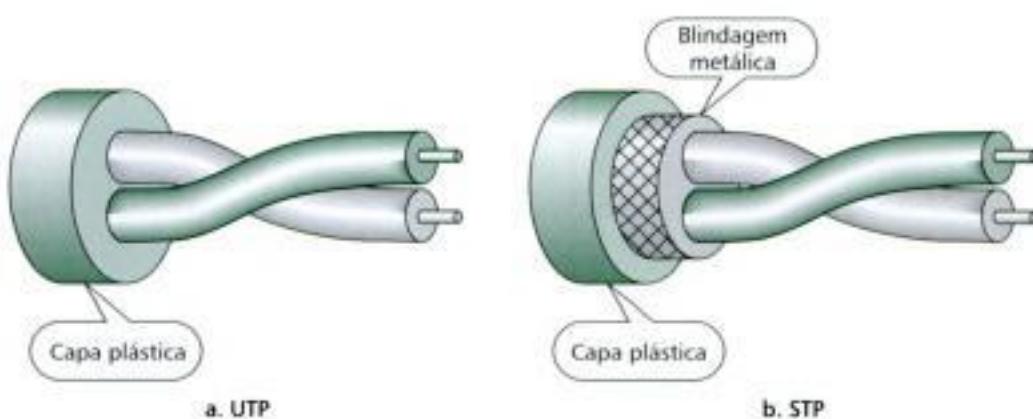


Figura 7.4 UTP e STP.

Categorias

A Electronics Industries Association (EIA) desenvolveu padrões para classificar cabos UTP em sete categorias (as categorias 6 e 7 ainda estão no estágio de minuta de padrão internacional*). As categorias são determinadas pela qualidade do cabo, onde 1 é mais baixa e 7 é a mais alta. Cada categoria EIA é adequada a um propósito específico. A Tabela 7.1 ilustra tais categorias.

TABELA 7.1 Categorias de cabos UTP

Categoria	Largura de banda	Taxa de transmissão	Digital/Analógico	Aplicação
1	muito baixa	< 100 kbps	Analógico	Telefone
2	< 2 MHz	2 Mbps	Analógico/digital	Linhas T-1
3	16 MHz	10 Mbps	Digital	LANs
4	20 MHz	20 Mbps	Digital	LANs
5	100 MHz	100 Mbps	Digital	LANs
6 (minuta)	200 MHz	200 Mbps	Digital	LANs
7 (minuta)	600 MHz	600 Mbps	Digital	LANs

* N. de R. T.: A categoria 6 não era um padrão no ano da publicação original do livro em inglês (2001). Hoje, ela é um padrão *de facto*.

Conectores

O conector UTP mais comum é o **RJ45** (RJ representa Registered Jack), ele aparece ilustrado na Figura 7.5. O RJ45 macho é um conector munido de um guia de encaixe, significando que ele pode ser inserido somente numa posição no RJ45 fêmea.

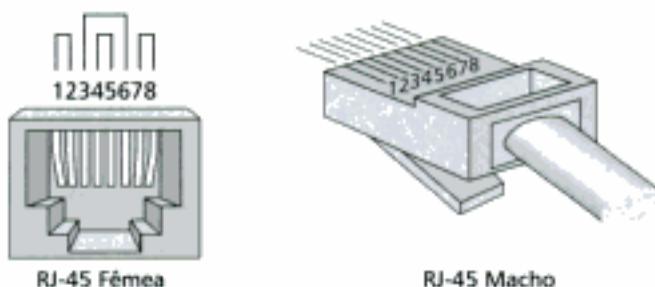


Figura 7.5 Conectores UTP.

Performance

Uma maneira de medir a *performance* de um cabo de pares trançados é comparar a atenuação *versus* freqüência e distância. Um cabo de pares trançados possui uma banda de freqüência muito ampla, assim, permite a passagem de freqüências numa faixa muito grande. Entretanto, a Figura 7.6 mostra que quando a freqüência aumenta, a atenuação (medida em decibel – dB), aumenta bruscamente por volta de 100 kHz. Note que a bitola (gauge*) dá a medida da espessura do fio.

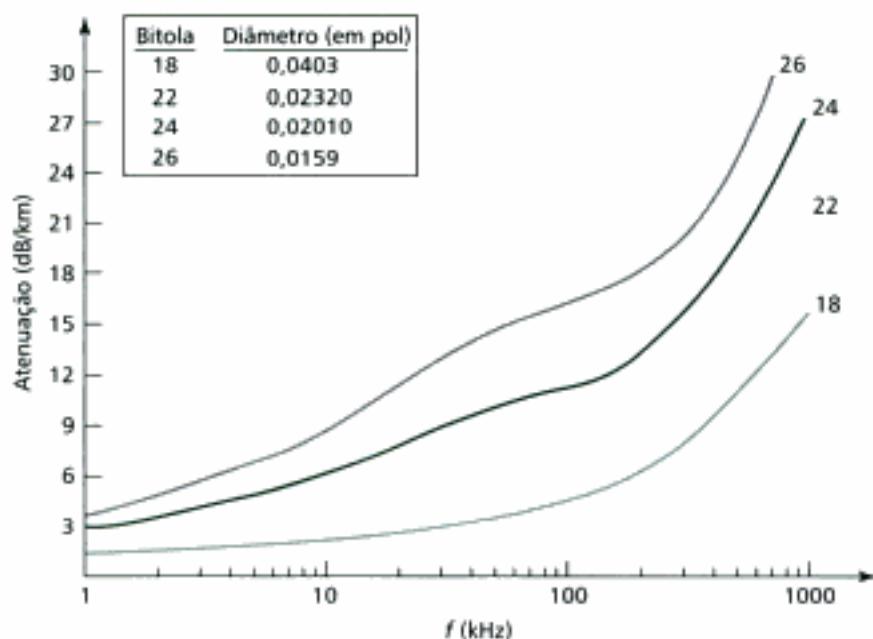


Figura 7.6 Performance UTP.

Aplicações

O cabo par trançado é utilizado em linhas telefônicas como meios físicos dos canais de voz e de dados. A conexão local do assinante, linha que conecta um assinante à central telefônica de uma empresa de telefonia, é realizada normalmente através do cabo par trançado. Examinaremos as redes telefônicas no Capítulo 8.

* N. de R. T.: O gauge é uma medida americana estabelecida no AWG (American Wire Gauge). O Brasil não segue essa norma.

As linhas DSL, utilizadas pelas companhias telefônicas para proporcionar conexões em banda larga usam a vasta capacidade de transmissão dos cabos UTP. Discutiremos cuidadosamente a tecnologia DSL no Capítulo 9.

As redes locais (LANs), tais como a 10BaseT e 100BaseT, também utilizam cabos UTP. As redes LAN serão abordadas no Capítulo 14.

Cabo Coaxial

Um cabo coaxial (*o coax*) transporta sinais em faixas de frequência superiores às do cabo UTP. Em parte, isso se deve aos meios serem construídos de forma muito diferente. Em vez de ter um par de fios, um coax tem um condutor no núcleo central (geralmente de cobre) encerrado por uma malha condutora externa de fios trançados e todo o conjunto é envolvido por um revestimento isolante. A malha serve tanto para blindagem contra ruídos como o segundo condutor, o que completa o circuito. Como dissemos, essa malha é envolvida pelo revestimento isolante e todo o cabo é protegido por uma cobertura plástica (veja Figura 7.7).

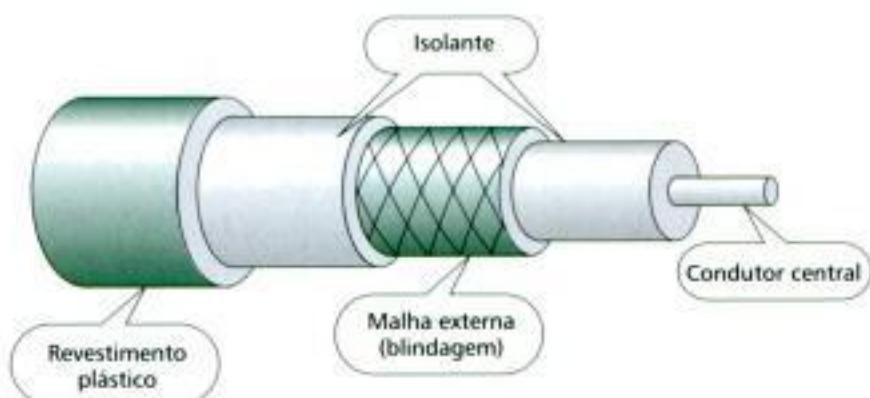


Figura 7.7 Cabo coaxial.

Padrões de Cabos Coaxiais

Os cabos coaxiais são categorizados através do número RG (Radio Government). Cada número RG denota um conjunto único de especificações físicas, incluindo a bitola do condutor interno no núcleo, a espessura e o tipo de isolamento interno, a construção da blindagem e o tamanho/tipo do revestimento plástico externo. Cada tipo de cabo, definido pelo número RG, é adaptado para uma função específica, conforme Tabela 7.2.

TABELA 7.2 Categorias de cabos coaxiais

Categoria	Impedância	Aplicação
RG-59	75 Ω	TV a Cabo
RG-58	50 Ω	<i>Thin Ethernet</i>
RG-11	50 Ω	<i>Thick Ethernet</i>

Conectores para Cabo Coaxial

Obviamente, os conectores para cabos coaxiais são diferentes dos conectores para cabo par trançado. O tipo mais comum de conector utilizado hoje em dia é o Bayonet-Neil-Concelman, ou simplesmente, conector BNC. A Figura 7.8 ilustra os três tipos freqüentemente encontrados: o conector BNC, o T-BNC e o terminador BNC.

O **conector BNC** é utilizado para conectar a extremidade de um cabo coaxial a um dispositivo, tal como um aparelho de TV. O T-BNC foi bastante utilizado em redes Ethernet (veja Capítulo 14) para dar prosseguimento ao cabeamento coaxial de modo a interligar um computador com outros dispositivos. O terminador BNC é utilizado numa das extremidades do cabo para realizar a reflexão do sinal.

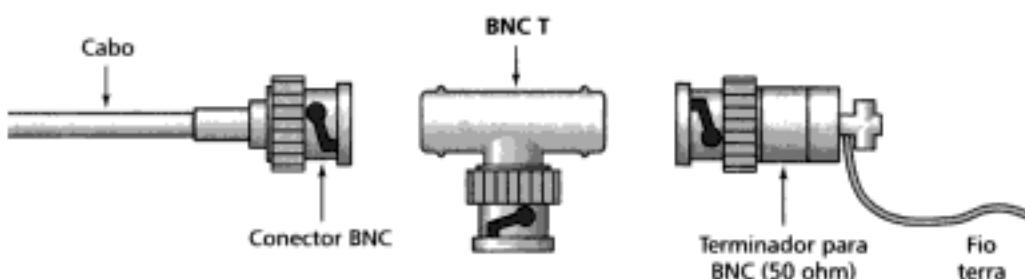


Figura 7.8 Conectores BNC.

Performance

Como fizemos para o cabo UTP, podemos medir a *performance* de um cabo coaxial. Na Figura 7.9, note que a atenuação é maior em cabos coaxiais do que em cabos UTP. Noutras palavras, embora o cabo coaxial disponha de uma largura de banda maior, o sinal enfraquece muito rapidamente, exigindo o uso freqüente de repetidores.

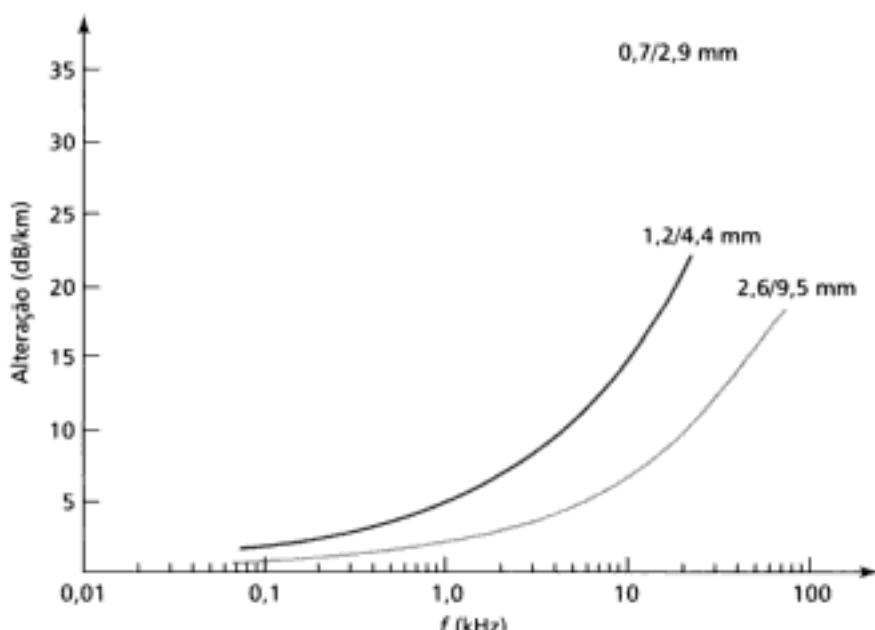


Figura 7.9 Performance do cabo coaxial.

Aplicações

O cabo coaxial encontrou as primeiras aplicações nas redes telefônicas analógicas onde uma única rede coaxial era capaz de transportar 10.000 sinais de voz. Mais recentemente, ele foi utilizado em redes telefônicas digitais sendo que um único cabo coaxial era capaz de transportar dados em taxas de até 600 Mbps. Entretanto, ultimamente o cabo coaxial tem sido sistematicamente substituído pelas fibras ópticas nas redes telefônicas.

As transmissões de TV a cabo (CATV) também utilizam cabos coaxiais. Na CATV tradicional, a rede inteira usa cabo coaxial. Porém, hoje também existe uma tendência de substituí-lo pela fibra óptica, a diferença é que ainda há uma certa preferência pelo cabo coaxial próximo às residências dos consumidores finais. A CATV usa o cabo coaxial RG-59. Podemos dizer que o estágio atual das redes CATV é formado por redes híbridas.

Outra aplicação muito comum do cabo coaxial é nas LANs Ethernet (veja Capítulo 14). Devido à enorme largura de banda do *coax* e, consequentemente, às altas taxas de transmissão permitidas nele, o cabo coaxial foi escolhido para transmitir dados nas redes Ethernet padrão. O padrão dessas redes é o 10Base2 ou o Thin Ethernet que utiliza o cabo coaxial fino RG-58 com conectores BNC para transmitir dados a 10 Mbps em distâncias até 185 m (sem repetidores). O padrão 10Base5 ou Thick Ethernet usa o cabo RG-11 (cabo coaxial grosso) para transmitir dados a 10 Mbps em distância até 500 m (sem repetidores). As redes 10Base5 usam conectores especiais.

Fibra Óptica

Uma fibra óptica é feita de vidro ou plástico e transmite sinais na forma de pulsos de luz. Para compreender o princípio de funcionamento da fibra óptica precisamos explorar primeiramente alguns aspectos relacionados à natureza da luz.

Raios luminosos viajam em linha reta quando se movem dentro de um mesmo meio. Se um raio de luz estiver viajando dentro de um meio e encontrar uma interface dividindo o meio onde está com outro (mais ou menos denso) pode mudar drasticamente a direção de propagação. A Figura 7.10 ilustra a mudança na direção de propagação de um raio de luz provocada pela mudança da densidade do meio. Na figura o raio propaga de um meio mais denso para outro menos denso.

A figura também mostra que, se o **ângulo de incidência** (o ângulo formado entre o raio e a linha perpendicular traçada a partir da interface entre as duas superfícies) for menor que o **ângulo crítico**, o raio sofre **refração** e passa a se mover numa nova direção mais próxima à superfície. Se o ângulo de incidência for exatamente igual ao ângulo crítico, o raio de luz viaja rente à superfície. Se o ângulo de incidência for maior que o ângulo crítico, o raio sofre **reflexão total**, muda novamente a direção, mas permanece dentro do meio onde estava viajando. O ângulo crítico é uma propriedade do meio óptico, logo o valor do ângulo depende do meio em questão.

Fibras ópticas usam reflexão total para guiar um sinal de luz através de um canal. Um **núcleo** de vidro ou plástico é encerrado por uma **casca** menos densa de vidro ou plástico. A diferença de densidade dos dois materiais deve ser tal que um feixe de luz movendo-se através do núcleo seja refletido de volta pela casca em vez de sofrer refração dentro dela (veja Figura 7.11).

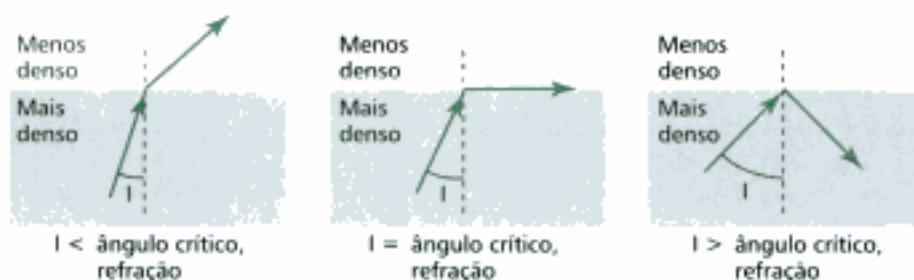


Figura 7.10 Desvio do raio luminoso.

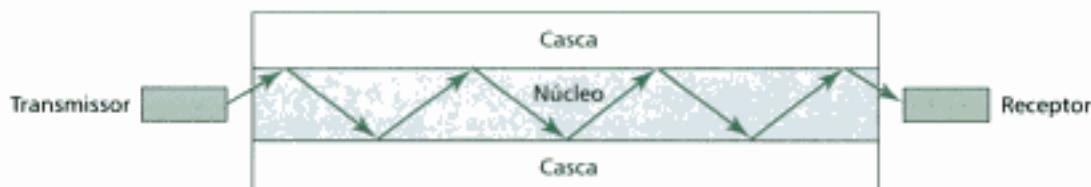


Figura 7.11 Fibra óptica.

Modos de Propagação

A tecnologia atual de fabricação de fibras ópticas suporta dois tipos de modos de propagação de luz dentro de fibras ópticas, a saber: monomodo e multimodo. Cada um desses tipos requer fibras

ópticas de diferentes características físicas. Ainda, as fibras multimodo podem ser implementadas de duas maneiras: índice degrau (*step-index*) ou índice gradual (*graded-index*), veja Figura 7.12.

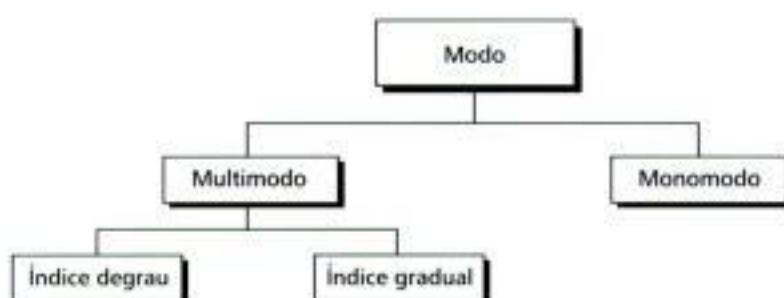


Figura 7.12 Modos de propagação.

Multimodo O nome multimodo decorre também do fato de que múltiplos feixes de luz oriundos de uma fonte de luz movem-se através do núcleo por caminhos diferentes. O modo como estes feixes se movem dentro do cabo, depende da estrutura do núcleo, como mostra a Figura 7.13.

Na **fibra multimodo índice degrau**, a densidade do núcleo permanece constante do centro até a borda da interface núcleo/casca. Um feixe de luz move-se em linha reta através desse meio com densidade constante até atingir a interface núcleo/casca. Na interface, ocorre uma mudança abrupta para uma densidade menor que altera o ângulo do movimento do feixe. O termo *índice degrau* refere-se a essa mudança abrupta na densidade do meio.

Um segundo tipo de fibra, denominada **fibra multimodo índice gradual**, diminui o nível de distorção do sinal provocado pelo movimento através do cabo. A palavra *índice* refere-se ao índice de refração do meio. Como vimos acima, o índice de refração está relacionado à densidade do meio. Uma fibra com índice gradual é aquela onde o índice de refração varia gradualmente. A densidade é mais alta no centro do núcleo e vai diminuindo gradualmente até atingir um valor mínimo na interface núcleo/casca. A Figura 7.13 mostra o impacto dessa densidade variável na propagação do feixe de luz.

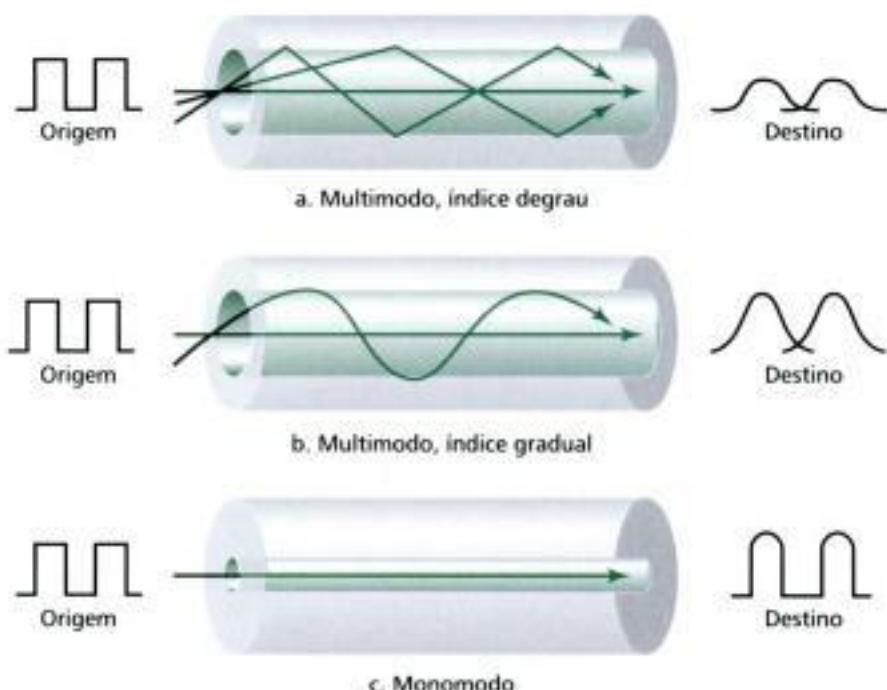


Figura 7.13 Modos.

Monomodo As fibras monomodo são construídas com índice em degrau e uma fonte de luz altamente focada que limita o feixe a pequenos ângulos de abertura, chegando a propagar o feixe na horizontal. As **fibras monomodo** são fabricadas com diâmetros ainda menores que as fibras multimodo e com uma densidade (índice de refração) substancialmente menor. A diminuição na densidade resulta num ângulo crítico muito próximo de 90°, provocando uma propagação quase horizontal do feixe de luz. Nesse caso, a propagação de diferentes feixes dentro do núcleo é bastante reduzida e podemos dizer que todos os feixes procuram a direção de propagação horizontal. Isso reflete na diminuição dos atrasos (*delays*) de propagação. Todos os feixes chegam ao destino juntos onde são recombinados, gerando relativamente pouca distorção no sinal (veja a Figura 7.13).

Tamanho das Fibras

As fibras ópticas são definidas pela razão entre o diâmetro do núcleo e o diâmetro da casca, ambas expressas em micrômetros. As dimensões mais comuns são mostradas na Tabela 7.3. O último tamanho listado na tabela é o único para a fibra monomodo.

TABELA 7.3 Tipos de fibras

<i>Tipo</i>	<i>Núcleo (μm)</i>	<i>Casca</i>	<i>Modo</i>
50/125	50	125	Multimodo, índice gradual
62,5/125	62,5	125	Multimodo, índice gradual
100/125	100	125	Multimodo, índice gradual
7/125	7	125	Monomodo

Composição do Cabo

A Figura 7.14 mostra a composição típica de um cabo de fibra óptica.

A jaqueta externa é feita de PVC ou Teflon. Dentro da jaqueta são adicionados fios de Kevlar para fortalecer o cabo. O Kevlar é um material resistente utilizado na fabricação de roupas à prova de balas. No nível abaixo do Kevlar encontramos outra cobertura plástica que serve para proteger a fibra contra choques. A fibra propriamente dita aparece no centro do cabo e consiste da casca e do núcleo.

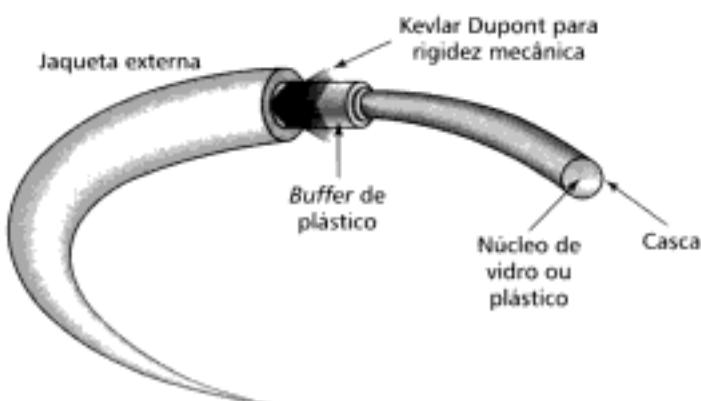


Figura 7.14 Partes da fibra.

Conectores para Cabos de Fibra Óptica

Os cabos de fibra óptica utilizam diferentes tipos de conectores, como mostra a Figura 7.15.

O **conector SC (subscriber channel)** é utilizado na TV a cabo. Ele utiliza um sistema de travamento tipo *push/pull*. O **conector ST (straight-tip)** é utilizado na conectorização dos cabos dispositivos de redes. Ele utiliza um sistema de travamento tipo balonete e é mais confiável que o SC. O conector **MT-RJ** é um novo tipo de conector com as mesmas dimensões do RJ45.

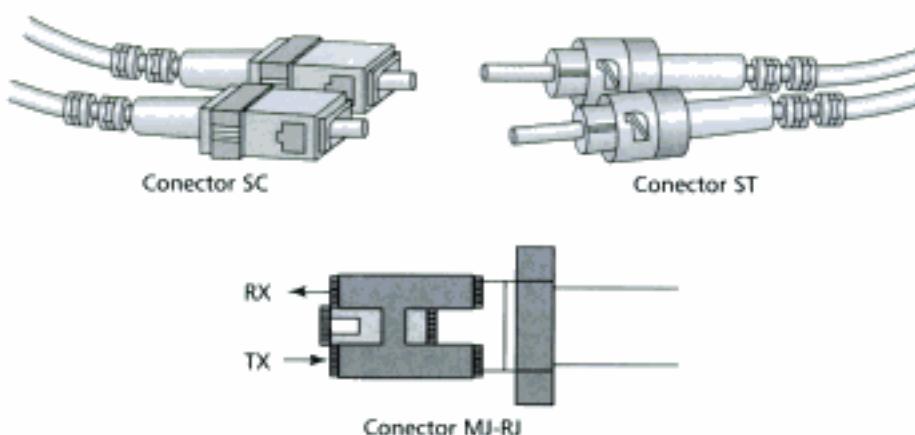


Figura 7.15 Conectores para cabos de fibra óptica.

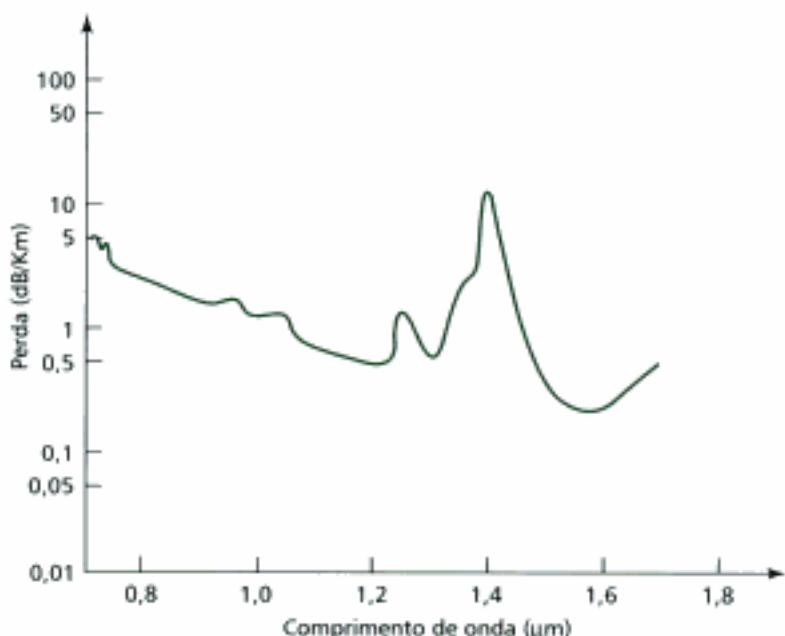


Figura 7.16 Performance da fibra óptica.

Performance

Um gráfico da medição da atenuação *versus* o comprimento de onda da luz mostra um fenômeno muito interessante de um cabo de fibra óptica. A atenuação é mais plana que os casos dos cabos par trançado e coaxial. A *performance* é tal que precisamos muito menos repetidores para replicar o sinal (cerca de 10 vezes menos) quando estamos utilizando um cabo de fibra óptica.

Aplicações

O cabo de fibra óptica é freqüentemente utilizado nos *backbones* de redes devido à relação entre a largura de banda muito alta e o custo efetivo. Hoje em dia, utilizando WDM, podemos transferir dados a taxas de 1600 Gbps. A rede SONET que discutiremos no Capítulo 9 utiliza fibras ópticas como *backbones*.

Algumas empresas de TV a cabo utilizam uma combinação de fibra óptica e cabo coaxial, criando assim, uma rede híbrida. As fibras ópticas desempenham papel de *backbones* estruturais enquanto que os cabos coaxiais proporcionam a conexão com os assinantes. Os custos não justificam levar fibras ópticas até o usuário final (o assinante), porque eles não necessitam de tanta banda.

As redes LANs, tais como a 100BaseFX (Fast Ethernet) e 1000BaseX, também utilizam cabos de fibra óptica.

Vantagens e Desvantagens das Fibras Ópticas

Vantagens Dentre as vantagens dos cabos de fibra óptica sobre os cabos metálicos (pares trançados ou coaxial) podemos citar:

- **Largura de banda.** As fibras ópticas suportam larguras de banda muito maiores que a maioria das aplicações de hoje necessitam. Assim, são fantasticamente superiores aos cabos par trançado e coaxial. Atualmente, as taxas de dados e a largura de banda disponíveis nas fibras ópticas estão limitadas não pelo meio, mas pela tecnologia de geração e recepção dos sinais.
- **Atenuação.** A distância de transmissão utilizando fibra óptica é significativamente superior à distância alcançada por qualquer um dos meios metálicos. Em algumas aplicações, um sinal pode viajar numa fibra óptica aproximadamente 50 km sem requerer regeneração, enquanto que num par trançado ou coaxial necessitamos introduzir repetidores a cada 5 km.
- **Imunidade à interferência eletromagnética.** Ruídos eletromagnéticos não podem afetar as transmissões nos cabos de fibra óptica.
- **Resistência à corrosão dos materiais.** Vidro é um material muito mais resistente que outros materiais corrosivos, como o cobre.
- **Peso.** Fibras ópticas são muito mais leves que os cabos de cobre.
- **Imune às derivações.** Fibras ópticas não permitem que os sinais que viajam no núcleo sejam bisbilhoteados através de emendas ou derivações no cabo. Os cabos de cobre, além de permitirem emendas, irradiam facilmente como antenas, por isso são menos seguros.

Desvantagens Aqui estão as principais desvantagens dos cabos de fibra óptica:

- **Instalação/Manutenção.** Os cabos de fibra óptica são tecnologia relativamente nova, se comparado aos cabos metálicos. Por isso, há problemas com os custos de instalação/manutenção e a de mão-de-obra especializada que ainda não está disponível em todos os lugares.
- **Unidirecional.** A propagação da luz é unidirecional. Como na maioria das aplicações a comunicação é *full-duplex*, os cabos de fibra óptica necessitam de duas fibras, uma para transmitir o sinal ótico, outra para receber a resposta.
- **Custo.** Tanto o cabo quanto as interfaces ópticas são relativamente mais caras que nos outros tipos de meios. Se a demanda por banda não for muito grande, freqüentemente o uso das fibras ópticas não é justificável.

7.2 TRANSMISSÃO SEM FIOS (WIRELESS)

Numa **transmissão sem fios** as ondas eletromagnéticas viajam sem utilizar um meio guia orientador como suporte físico. Este tipo de comunicação é freqüentemente conhecido como **comunicação sem fios** ou **wireless**. Nesta transmissão, os sinais de radiodifusão viajam através do ar, ficando acessíveis a quem quer que disponha de um dispositivo capaz de recebê-los.

A Figura 7.17 mostra parte do espectro eletromagnético, cobrindo a faixa de 3 kHz a 900 THz, usada na comunicação wireless.

Os sinais não guiados podem viajar da fonte ao destino de diversas maneiras possíveis. Podemos citar as propagações no solo, ionosférica e direcionada, como mostra a Figura 7.18.

Na **propagação no solo**, as ondas de rádio viajam através da porção mais baixa da atmosfera, podemos dizer, quase na superfície da Terra. Estes sinais de baixa freqüência irradiam em todas as direções, partindo da antena transmissora, seguindo a curvatura do planeta. As distâncias atingidas dependem da potência do sinal transmitido: quanto maior a potência, maior a distância alcançada. Na **propagação ionosférica** ondas de rádio de alta freqüência são irradiadas em direção à ionosfera (camada da atmosfera onde as partículas estão ionizadas) onde elas são

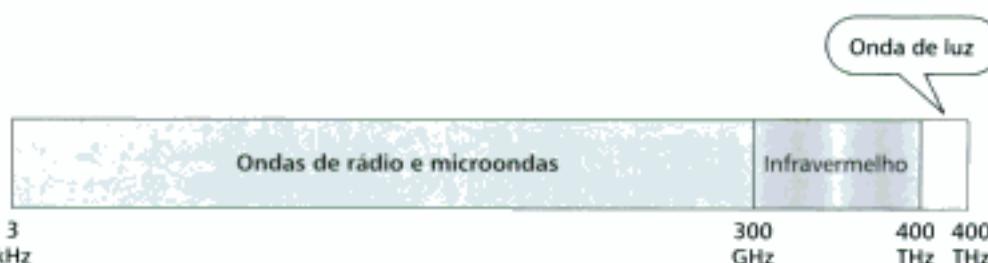


Figura 7.17 Espectro eletromagnético para comunicação wireless.

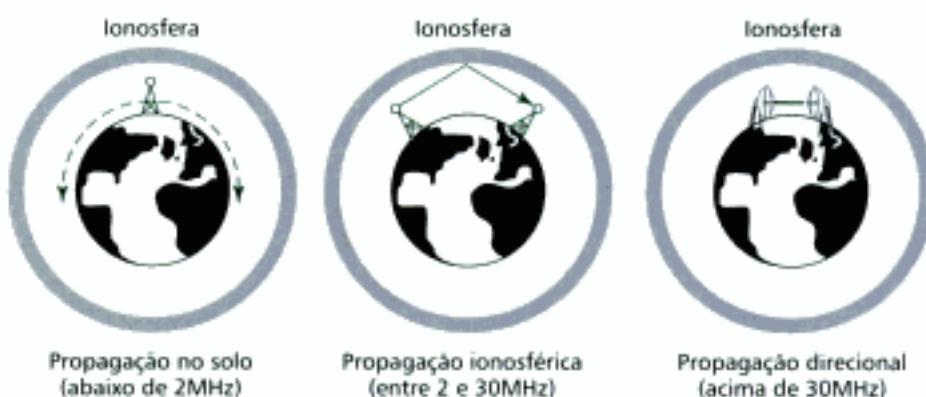


Figura 7.18 Métodos de propagação

refletidas de volta à superfície. Este tipo de transmissão permite atingir grandes distâncias com relativamente pouca potência. Na **propagação direcional** sinais de freqüências muito altas são transmitidas em linha reta, de antena a antena. As antenas devem estar direcionadas face a face e suficientemente próximas para que a curvatura da Terra não afete a transmissão. A propagação direcionada é enganosa porque as transmissões de rádio não podem ser completamente focadas.

A faixa do espectro eletromagnético definida como ondas de rádio e microondas é dividida em oito faixas, denominadas *bandas*, cada qual regulamentada por autoridades governamentais. Estas bandas estendem-se das freqüências muito baixas (VLF) até freqüências extremamente altas (EHF). A Tabela 7.4 lista estas bandas, as faixas correspondentes, método de propagação e algumas aplicações.

Dividimos as transmissões wireless em três grandes grupos: ondas de rádio, microondas e ondas de infravermelho. Veja a Figura 7.19.

TABELA 7.4 Bandas

Banda	Faixa	Propagação	Aplicação
VLF	3–30 KHz	Solo	Rádio navegação
LF	30–300 KHz	Solo	Orientação de rádio para aviadores
MF	300 KHz–3 MHz	Ionosférica	Rádio AM
HF	3–30 MHz	Ionosférica	Faixa Cidadão (CB), Comunicação aérea/marítima
VHF	30–300 MHz	Ionosférica e direcional	TV VHF, rádio FM
UHF	300 MHz–3 GHz	Direcional	TV UHF, celulares, paging, satélite
SHF	3–30 GHz	Direcional	Comunicação via satélite
EHF	30–300 GHz	Direcional	Radar, satélite



Figura 7.19 Transmissão de ondas sem fios.

Ondas de Rádio

Embora não exista nenhuma demarcação clara entre ondas de rádio e microondas, isto é, onde termina uma e começa a outra, as ondas eletromagnéticas cobrindo a faixa de 3 kHz a 1 GHz são normalmente denominadas **ondas de rádio**; as ondas na faixa de freqüência entre 1 e 300 GHz são as conhecidas **microondas**. Entretanto, o comportamento das ondas, em vez das freqüências, é um critério melhor de classificação.

Ondas de rádio, na sua grande maioria, são omnidirecionais. Quando uma antena transmite ondas de rádio elas propagam em todas as direções. Isto significa que as antenas transmissora e receptora não precisam estar alinhadas. Uma antena transmissora emite ondas que podem ser captadas por qualquer antena receptora. A propriedade de omnidirecionalidade é uma grande desvantagem também: as ondas de rádio transmitidas por uma antena estão susceptíveis às interferências provocadas por outra antena que estiver emitindo na mesma freqüência ou banda.

As ondas de rádio, particularmente aquelas que viajam até a ionosfera, podem percorrer longas distâncias com relativamente pouca potência de sinal. Isto fez das ondas de rádio as candidatas imediatas para cobrir longas distâncias na radiodifusão e gerou, por exemplo, as rádios AM.

As ondas de rádio, particularmente aquelas de baixa ou média freqüências, podem "penetrar em paredes". Esta característica pode ser encarada tanto como uma vantagem quanto uma desvantagem. A vantagem é que uma rádio AM pode ser sintonizada dentro de uma construção. E a desvantagem é que não é possível isolar a comunicação apenas interna ou externamente a uma construção. As ondas de rádio ocupam uma banda relativamente estreita, apenas 1 GHz, se comparada à banda de microondas. Quando dividimos estas bandas em bandas menores, as bandas laterais também são estreitadas para permitir a comunicação digital de dados em taxas pequenas.

Quase a banda inteira é regulada pelas autoridades governamentais (p. ex.: o FCC nos Estados Unidos). Para utilizar qualquer faixa da banda é necessário obter permissão dessas autoridades.

Antena Omnidirecional

Conforme vimos acima, as ondas de rádio usam **antenas omnidirecionais** para transmitir sinais em todas as direções. Podemos encontrar diversos tipos de antenas desse tipo baseados no comprimento de onda, intensidade do sinal e o propósito da transmissão. A Figura 7.20 ilustra o funcionamento de uma antena omnidirecional.

Aplicações

As características de omnidirecionalidade das ondas de rádio as tornam úteis na multidifusão (*multicasting*), onde existem um transmissor e muitos receptores. As rádios AM e FM, televisão, raiodionavegação e telefones sem fio são exemplos de *multicasting*.

Ondas de rádio são utilizadas na comunicação multidifusão (*multicasting*), tal como rádio e televisão.

Microondas

As microondas são as ondas eletromagnéticas cujas freqüências estão compreendidas entre 1 e 300 GHz.

Microondas são sempre unidirecionais. Quando uma antena transmite microondas ela precisa estar focada com a antena receptora para que a comunicação aconteça com inteligibilidade.



Figura 7.20 Antenas omnidirecionais.

Essa propriedade de unidirecionalidade tem uma vantagem óbvia: somente o par de antenas que estiverem alinhadas podem se comunicar. Além disso, essa propriedade impede que um par de antenas provoque interferência eletromagnética em outra.

Como a transmissão via microondas é direcional, o par de torres onde são montadas as antenas precisa estar devidamente alinhado. Assim, torres que estiverem muito afastadas entre si não podem encontrar obstáculos para a transmissão no meio do caminho. Esse problema geralmente é resolvido colocando as torres em lugares altos. A curvatura da Terra é outro problema para a transmissão das microondas. Quando a comunicação deve ser realizada entre dois pontos geograficamente afastados são necessários repetidores para contornar o efeito da curvatura do planeta ou para regenerar a intensidade do sinal.

As microondas têm freqüências altíssimas e, por isso, não tem um poder de penetração em paredes e obstáculos muito grande. Esta característica pode ser uma desvantagem se os receptores estiverem dentro de construções.

Vimos que a largura de banda das microondas é relativamente alta, cerca de 299 GHz. Desse modo, é possível dividi-la em faixas relativamente grandes e possibilitar a comunicação de dados em taxas elevadas.

Para usar uma faixa do espectro de microondas é necessária permissão das autoridades governamentais competentes.

Antena Unidirecional

As microondas necessitam de **antenas unidirecionais** para enviar sinais numa única direção. Existem dois tipos de antenas para comunicação via microondas: prato parabólico e corneta (veja Figura 7.21).

Uma **antena prato parabólico** foi baseada construtivamente na geometria de uma parábola. De acordo com a física das lentes e espelhos parabólicos, cada raio paralelo à linha de simetria da parábola é refletido num ângulo tal que todas as linhas paralelas se interceptam num ponto comum, denominado foco. Assim, o prato parabólico funciona como um "funil" para as ondas eletromagnéticas. Ele as apanha e direciona para um ponto comum, o foco. Dessa maneira, a maior parte dos sinais que chega ao prato parabólico é recuperado e direcionado a um ponto comum (o foco) onde é montado o receptor.

Uma **antena corneta** é um captador direcional que mais se parece com uma concha gigantesca. As transmissões são realizadas em multidifusão num tronco (guia de onda) e são defletidas numa série de feixes de ondas paralelas pela peça curvada na extremidade da antena. As transmis-

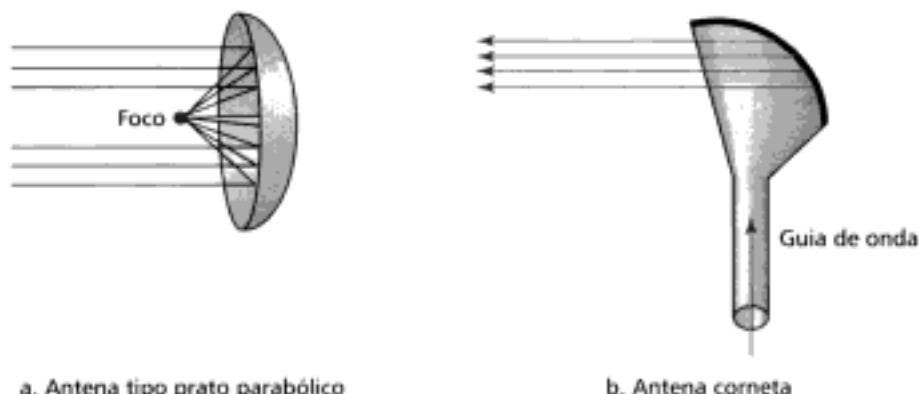


Figura 7.21 Antenas unidirecionais.

sões são recebidas por outra antena do mesmo tipo, mais especificamente na concha dela, de maneira bastante similar ao prato parabólico e são defletidos para dentro do guia de ondas do tronco.

Aplicações

Devido às propriedades de unidirecionalidade, as microondas são muito úteis em comunicações *unicasting* (um a um) entre dois dispositivos transmissor e receptor. Elas são bastante utilizadas em telefonia celular (Capítulo 17), rede de satélites (Capítulo 17) e nas wireless LANs (Capítulo 15).

As microondas são muito úteis na comunicação *unicast* como em telefones celulares, redes de satélites e nas wireless LANs.

Infravermelho

Os sinais de infravermelho têm freqüências compreendidas entre 300 GHz e 400 THz (comprimentos de onda de 1 mm a 770 nm) e são utilizados apenas para comunicações em curto alcance. As freqüências dos sinais infravermelho são tão elevadas que eles não podem penetrar em obstáculos como paredes. Esta aparente desvantagem previne contra interferência entre dois sistemas infravermelho que sejam montados até mesmo num mesmo prédio, mas em ambientes (salas) diferentes. Por exemplo, quando utilizamos o controle remoto infravermelho de algum aparelho eletrônico em nossa residência, não provocamos interferência nos controles remotos dos aparelhos dos nossos vizinhos. Entretanto, essa mesma característica inviabiliza o uso do infravermelho para comunicações em longas distâncias. Além disso, não podemos utilizar ondas eletromagnéticas na região do infravermelho fora de uma construção porque os raios do sol são compostos, em parte, de ondas infravermelhas que fatalmente iriam interferir nas comunicações.

Aplicações

A banda do infravermelho (quase 400 THz) possui um potencial excelente para transmissão de dados. Por exemplo, imagine toda essa banda disponível para comunicação de dados dentro de uma empresa, as taxas de transmissão seriam muito superiores dos níveis atuais. A Infrared Data Association (IrDA) é uma associação que dissemina o uso das ondas infravermelhas e tem desenvolvido muitos padrões para o uso desses sinais na comunicação de dados entre dispositivos, tal como teclado, mouse, PCs e impressoras. Por exemplo, alguns fabricantes disponibilizam portas especiais denominadas **IrDA** que permitem a conexão de teclado sem fio para comunicar com um PC. O padrão originalmente desenvolvido definia uma taxa de transmissão de 75 kbps para uma distância de até 8 m. O padrão mais recente define taxas de transmissão de 4 Mbps.

Sinais infravermelho definidos pela IrDA são transmitidos direcionalmente; a porta IrDA no teclado precisa apontar para o PC para que ocorra a transmissão.

Sinais infravermelhos só podem ser utilizados para comunicação a curtas distâncias, em áreas fechadas e utilizando propagação direcionada.

7.3 TERMOS-CHAVE

Ângulo crítico	Microonda
Ângulo de incidência	MT-RJ
Antena corneta	Núcleo
Antena de prato parabólico	Número RG
Antena omnidirecional	Onda de rádio
Antena unidirecional	Onda infravermelha
Cabo coaxial	Par trançado blindado (<i>Shielded Twisted-pair – STP</i>)
Cabo de fibra óptica	Par trançado sem blindagem (<i>Unshielded Twisted-pair – UTP</i>)
Cabo de pares trançados	Porta IrDA
Casca	Propagação direcional
Comunicação sem fio (<i>wireless</i>)	Propagação ionosférica
Conector BNC	Propagação no solo
Conector SC (<i>Subscriber-Channel</i>)	Reflexão
Conector ST (<i>Straight-Tip</i>)	Refração
Espectro eletromagnético	RJ45
Fibra monomodo	Transmissão guiada
Fibra óptica	Transmissão sem fio (<i>wireless</i>)
Fibra óptica índice degrau	
Fibra óptica índice gradual	
Meios de transmissão	

7.4 RESUMO

- Os meios de transmissão situam-se logo abaixo da camada física.
- Uma transmissão guiada proporciona um caminho físico entre os dispositivos transmissor e o receptor.
- Dentre os meios de transmissão que servem como guias de onda podemos citar: cabo par trançado, cabo coaxial e fibra óptica.
- O cabo par trançado consiste de dois fios de cobre isolados e trançados juntos. O ato de trançar os dois fios permite que cada fio sofra o mesmo nível de interferência do ruído ambiente.
- O cabo par trançado é utilizado nas linhas telefônicas para transmissão de voz e de dados.
- O cabo coaxial possui a seguinte configuração física (partindo do centro): um condutor central, um isolante em volta do condutor, uma malha condutora externa (blindagem), um isolante cobrindo a blindagem e uma capa plástica.
- Cabos coaxiais podem transportar sinais numa faixa de frequência superior ao cabo par trançado.
- Cabo coaxial é usado na TV a cabo e na rede Ethernet padrão.
- Cabos de fibra óptica são compostos de um núcleo de vidro ou plástico revestido por uma casca, tudo isso encerrado numa jaqueta externa.
- Cabos de fibra óptica transportam sinais na forma de luz. O sinal luminoso propaga por reflexão total internamente no núcleo.
- A transmissão via fibra óptica está se tornando incrivelmente comum devido à resistência aos ruídos, baixa atenuação e altas larguras de banda das fibras.
- A propagação dos sinais nas fibras ópticas pode ser multimodo (múltiplos feixes de uma única fonte de luz) ou monomodo (essencialmente um único feixe da fonte de luz).
- Numa fibra óptica índice degrau, a densidade do núcleo é constante e o feixe de luz muda abruptamente a direção de propagação na interface entre o núcleo e a casca.
- Numa fibra óptica índice gradual, a densidade do núcleo diminui com a distância a partir do centro. Isto provoca o encurvamento do sinal luminoso.
- Fibras ópticas são utilizadas nos *backbones* de redes, redes de TV a cabo e redes Fast Ethernet.
- Numa transmissão sem fios, o meio (usualmente o ar) transporta ondas eletromagnéticas sem utilizar um meio condutor físico especialmente montado para essa finalidade.
- Na comunicação *wireless*, os dados podem ser transportados através da propagação no solo, ionosfera e/ou direcional.

- A comunicação *wireless* pode ser classificada como ondas de rádio, microondas ou sinal infravermelhos.
- Ondas de rádio são omnidirecionais. A faixa de freqüências de rádio é controlada pelo governo.
- Microondas são unidirecionais, isto é, a propagação deve acontecer em linha reta. As microondas são utilizadas na telefonia

celular, via satélite e nas comunicações *wireless LAN*.

- A antena de prato parabólico e a antena corneta são utilizadas para transmissão e recepção de microondas.
- Sinais infravermelhos são utilizados nas comunicações a curta distância, tal como entre um PC e um dispositivo periférico.

7.5 PRATIQUE OS CONHECIMENTOS ADQUIRIDOS

Questões de Revisão

1. Um meio de transmissão faz parte da camada física? Por quê?
2. Cite os nomes das duas categorias de meios de transmissão.
3. Como uma transmissão guiada difere da transmissão sem fio?
4. Quais são os três tipos mais comuns de meios guias para comunicação de dados?
5. Que tipo de sinal é utilizado nas transmissões nos cabos par trançado e coaxial? Como ele difere do tipo de sinal no cabo de fibra óptica?
6. Cite uma aplicação de cada um dos três tipos de meios guias de transmissão.
7. Qual é a maior vantagem do cabo STP sobre o UTP?
8. Qual é o significado da palavra *trançado* no cabo par trançado?
9. Por que um cabo coaxial é superior a um cabo par trançado?
10. O que é reflexão?
11. Discuta os modos de propagação da luz dentro de canais ópticos.
12. Qual é o propósito da casca numa fibra óptica? Compare a densidade relativa da casca com a densidade do núcleo.
13. Cite as vantagens do cabo fibra óptica sobre os demais cabos.
14. Quais são as vantagens da fibra óptica como um meio de transmissão?
15. Cite as três maneiras dos dados propagarem numa transmissão *wireless*.
16. Cite uma aplicação para cada um dos meios de transmissão sem fios.
17. Como a propagação ionosférica difere da propagação direcional?
18. Qual é a diferença entre as ondas omnidirecionais e unidirecionais?
19. O que é uma porta IrDA?

Questões de Múltipla Escolha

20. Os meios de transmissão são usualmente classificados como _____.
 - a. Fixos e não fixos.
 - b. Guiados e sem fios.
 - c. Determinado e indeterminado.
 - d. Metálico e não metálico.
21. Os meios de transmissão estão próximos da camada _____.
 - a. Física
 - b. Rede
 - c. Transporte
 - d. Aplicação
22. Os cabos de categoria 1 UTP são frequentemente utilizados em redes _____.
 - a. Fast Ethernet
 - b. Ethernet padrão
23. Os conectores BNC são usados nos cabos _____.
 - a. UTP
 - b. STP
 - c. Coaxial
 - d. Fibra óptica
24. Um cabo _____ consiste de um condutor interno no núcleo e um segundo condutor externo na forma de malha de blindagem.
 - a. Par trançado
 - b. Coaxial
 - c. Fibra óptica
 - d. STP

25. Nas fibras ópticas, a fonte de sinal emite ondas de _____.
 a. Luz
 b. Rádio
 c. Infravermelho
 d. Freqüência muito baixa
26. Sinais de fumaça são um exemplo de comunicação _____.
 a. Através de um meio guia
 b. Sem fio
 c. Através de refração
 d. Através de um meio grande ou pequeno
27. Qual das seguintes opções utiliza comunicação guiada?
 a. Sistema de telefonia celular
 b. Sistema de telefonia fixa
 c. Comunicações via satélite
 d. Radiodifusão
28. Qual das seguintes opções não funciona como uma meio guia de onda?
 a. Cabo par trançado
 b. Cabo coaxial
 c. Cabo fibra óptica
 d. Atmosfera
29. Num ambiente repleto de dispositivos de alta tensão, o melhor meio de transmissão seria _____.
 a. Cabo par trançado
 b. Cabo coaxial
 c. Cabo fibra óptica
 d. A atmosfera
30. Qual é o maior fator que torna o cabo coaxial menos suscetível a ruídos que o par trançado?
 a. Condutor interno
 b. Diâmetro do cabo
 c. Condutor externo
 d. Material isolante
31. O número RG dá informações preciosas sobre _____.
 a. Pares trançados
 b. Cabos coaxiais
 c. Fibras ópticas
 d. Todas acima
32. Numa fibra óptica, o núcleo interno é _____ casca.
 a. Mais denso que a
 b. Menos denso que a
 c. Tem a mesma densidade da
 d. Nenhuma das anteriores
33. O núcleo interno de uma fibra óptica é composto de _____.
 a. Vidro ou plástico
- b. Cobre
 c. Bimetálico
 d. Líquido
34. Fibras ópticas, diferentemente dos meios metálicos, são altamente imunes a _____.
 a. Transmissão em alta freqüência
 b. Transmissão em baixa freqüência
 c. Interferência eletromagnética
 d. Refração
35. Quando um feixe de luz viaja através de um meio com duas densidades diferentes, se o ângulo de incidência for maior que o ângulo crítico, ocorre _____.
 a. Reflexão
 b. Refração
 c. Incidência
 d. Criticalidade
36. Quando o ângulo de incidência é _____ ângulo crítico, o feixe de luz aproxima-se da superfície da interface.
 a. Maior que o
 b. Menor que o
 c. Igual ao
 d. Nenhuma das anteriores
37. Numa propagação em _____, o feixe de luz propaga-se quase horizontalmente e o núcleo de baixa densidade possui um diâmetro pequeno, se comparado ao outros modos de propagação.
 a. Multimodo índice degrau
 b. Multimodo índice gradual
 c. Multimodo monomodo
 d. Monomodo
38. O método de propagação em _____ está sujeito às maiores distorções no sinal.
 a. Multimodo índice degrau
 b. Multimodo índice gradual
 c. Multimodo monomodo
 d. Monomodo
39. Numa propagação em _____, o núcleo possui densidade variável.
 a. Multimodo índice degrau
 b. Multimodo índice gradual
 c. Multimodo monomodo
 d. Monomodo
40. Quando nos referimos aos meios sem fios, normalmente estamos falando sobre _____.
 a. Fios metálicos
 b. Fios não metálicos

- c. O ar
d. Nenhuma das anteriores
41. A faixa de freqüências das ondas de rádio e de microondas vão de _____
a. 3 a 300 kHz
b. 300 kHz a 3 GHz
c. 3 kHz a 300 GHz
d. 3 kHz a 3000 GHz
42. Na propagação _____, ondas de rádio de baixa freqüência viajam próximas à superfície da Terra.
a. No solo
b. Ionosférica
c. Direcional
d. Espacial
43. As bandas VLF e LF propagam-se _____ nas comunicações.
a. No solo
- b. Ionosférica
c. Direcional
d. Espacial
- b. Ionosférica
c. Direcionalmente
d. Espacialmente
44. Uma antena de prato parabólico é uma antena _____.
a. Omnidirecional
b. Bidirecional
c. Unidirecional
d. Corneta
45. A _____ é uma associação que difunde o uso das ondas infravermelho.
a. IrDA
b. EIA
c. FCC
d. PUD

Exercícios

46. Um feixe de luz move-se de um meio a outro meio menos denso. O ângulo crítico vale 60° . Desenhe o percurso que a luz faz em ambos meios quando o ângulo de incidência vale:
a. 40°
b. 50°
c. 60°
d. 70°
e. 80°
47. Um cabo par trançado tem uma atenuação de 2 dB/km a 1 kHz. Qual é a atenuação para 20 km de cabo?
48. Como podemos inferir da Figura 7.6 que a largura de banda de um cabo par trançado está relacionada à distância?
49. Como podemos inferir da Figura 7.9 que a largura de banda de um cabo coaxial está relacionada à distância?
50. Como podemos inferir da Figura 7.16 que a largura de banda de uma fibra está relacionada à distância?
51. Se a velocidade da luz numa fibra vale 2×10^8 m/s, qual é a largura de banda de uma fibra que transmite sinais de luz com comprimentos de onda variando entre 1000 e 1500 nm sem perda significativa na intensidade do sinal?

Comutação de Circuitos e Redes de Telefonia

Toda vez que dispomos de vários dispositivos que necessitam comunicar-se, esbarramos no problema de como conectá-los de maneira a proporcionar uma comunicação eficiente entre dois dispositivos individuais. Uma solução é instalar uma **conexão ponto a ponto** entre cada par de dispositivos (topologia em malha) ou entre um dispositivo central e cada um dos outros dispositivos (topologia em estrela). Contudo, quando aplicados a redes muitas grandes, estes métodos são impraticáveis e imprevidentes. A quantidade e o comprimento dos *links* requerem demasiada infra-estrutura, sendo pouco atrativas do ponto de vista de custos. Além disso, a maioria dos *links* poderia ficar ociosa durante a maior parte do tempo. Tomemos um exemplo: imagine uma rede composta de seis dispositivos, denotados por A, B, C, D, E e F. Se o dispositivo A estabelecer uma conexão ponto a ponto com os dispositivos B, C, D, E e F, então, quando somente A e B estiverem conectados, os *links* conectando A e todos os demais dispositivos estariam ociosos e sendo desperdiçados.

Outras topologias que empregam conexões multiponto, tal como a barramento, são descartadas de imediato porque as distâncias e o número total de dispositivos aumentam além da capacidade dos meios e dos equipamentos utilizados.

Uma solução melhor é a comutação. Uma rede comutada consiste de uma série de nós de comutação, denominados **comutadores**. Os comutadores são dispositivos de *hardware* e/ou *software* capazes de estabelecer conexões temporárias entre dois ou mais dispositivos conectados ao comutador, e não entre si. Numa rede comutada, alguns desses nós são conectados aos dispositivos que devem comunicar-se. Outros são utilizados apenas como caminho ou rota.

Existem três tipos tradicionais de métodos de comutação, a saber: comutação de circuitos, comutação de pacotes e comutação de mensagens.

Neste capítulo, examinaremos a comutação de circuitos, a qual toma lugar na camada física do modelo OSI. Em seguida, abordaremos uma aplicação da comutação de circuitos, a rede de telefonia fixa.

8.1 COMUTAÇÃO DE CIRCUITOS

A **comutação de circuitos** estabelece uma conexão física direta entre dois dispositivos, tais como telefones ou computadores. Por exemplo, na Figura 8.1, em vez de estabelecermos conexões ponto a ponto entre os três telefones à esquerda (A, B e C) e os quatro telefones à direita (D, E, F e G), requerendo 12 *links* ao todo, podemos utilizar quatro nós de comutação e reduzir o número e o com-

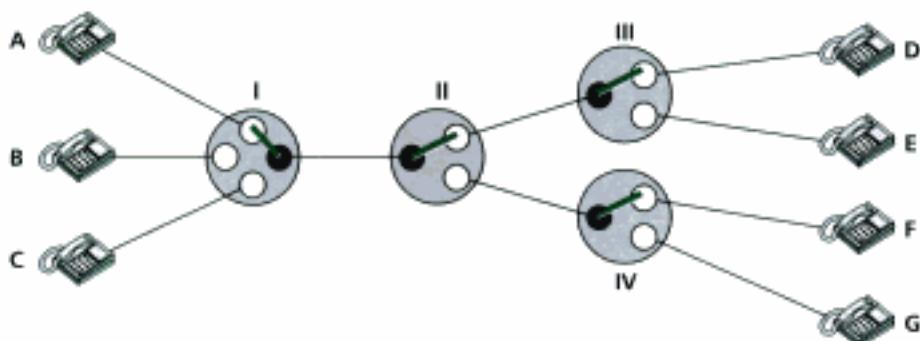


Figura 8.1 Rede de comutação de circuitos.

primento total dos *links*. Na Figura 8.1, o aparelho telefônico A está conectado através dos nós I, II e III ao aparelho D. Comutando a posição das chaves dentro dos nós, quaisquer telefones no lado esquerdo podem ser conectados aos telefones no lado direito.

Um circuito de comutação é um dispositivo com n entradas e m saídas que cria uma conexão temporária entre um *link* de entrada e um *link* de saída (veja a Figura 8.2). O número de entradas de um circuito não tem correlação com o número de saídas.

Um comutador n para n permite conectar n linhas entre si em modo *full-duplex*. Por exemplo, ele pode conectar n aparelhos telefônicos de tal modo que cada aparelho no lado esquerdo pode ser conectado a todos os demais, através da realimentação das linhas (veja Figura 8.3).

Hoje em dia, a comutação de circuitos pode ser encontrada em duas tecnologias: comutação por divisão de espaço ou comutação por divisão de tempo.

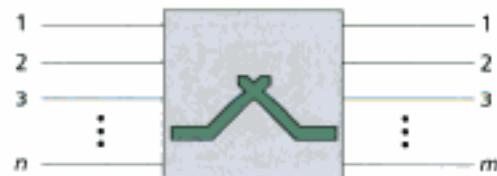


Figura 8.2 Um dispositivo de comutação.

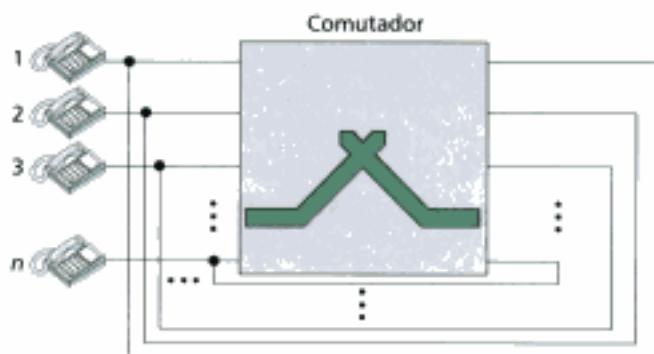


Figura 8.3 Um exemplo de comutação realimentada.

Comutação por Divisão de Espaço

Um **comutador matricial** conecta n entradas a m saídas, formando uma matriz e utilizando microchaves eletrônicas (transistores) em cada ponto de **cruzamento na matriz** (veja Figura 8.4). A maior limitação desta implementação é o número de pontos matriciais requeridos. Para conectar n entradas a m saídas utilizando um comutador desse tipo, são necessárias $n \times m$ microchaves. Por exemplo, para conectar 1000 entradas a 1000 saídas são necessários 1.000.000 de pon-

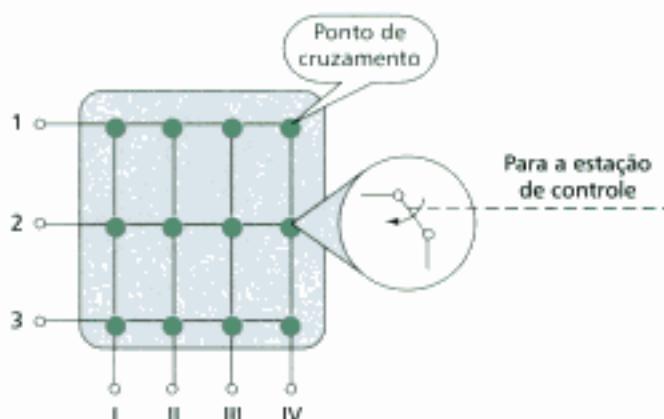


Figura 8.4 Um comutador matricial.

tos de cruzamento na matriz. Uma estrutura matricial com esse número de pontos é impraticável. Este tipo de comutação também é ineficiente porque, como mostram algumas estatísticas, menos que 25% dos pontos de cruzamento são utilizados num dado tempo. O restante dos pontos ficam ociosos.

Comutador Multiestágios

A solução para as limitações do comutador matricial é o **comutador multiestágios**, o qual combina comutadores matriciais distribuídos em estágios cascadeados. Na comutação multiestágios, os dispositivos de comunicação são interligados aos comutadores no primeiro estágio que, por sua vez, são ligados aos outros comutadores (veja a Figura 8.5).

A implementação de um comutador multiestágios depende do número de estágios e do número de comutadores requeridos ou desejados em cada estágio. Normalmente, os estágios intermediários possuem menos comutadores que o primeiro e último estágios. Por exemplo, imagine que desejamos fazer um comutador semelhante ao mostrado na Figura 8.5 realizar um trabalho de um comutador matricial 15 por 15. Assuma que vamos decidir utilizar uma implementação multiestágios formado por três estágios, utilizando três comutadores no primeiro e último estágios e dois comutadores no estágio intermediário. Como possuímos três comutadores no primeiro estágio e cada qual dispõe de cinco entradas, o número total de entradas combinadas nos comutadores será de $5 \times 3 = 15$ entradas.

Seguindo o raciocínio, temos três comutadores no primeiro estágio que devem ser conectados aos comutadores no lado da saída. Cada comutador do primeiro estágio deve possuir uma saída conectada a uma das entradas dos comutadores intermediários. Havendo dois comutadores intermediários, cada comutador do primeiro estágio deve possuir duas saídas, uma para cada entrada dos comutadores intermediários. Além disso, o comutador do terceiro estágio deve receber entradas de cada um dos dois comutadores intermediários e, desse modo, ter dois comutadores intermediários significa duas entradas em cada comutador no estágio final. Os comuta-

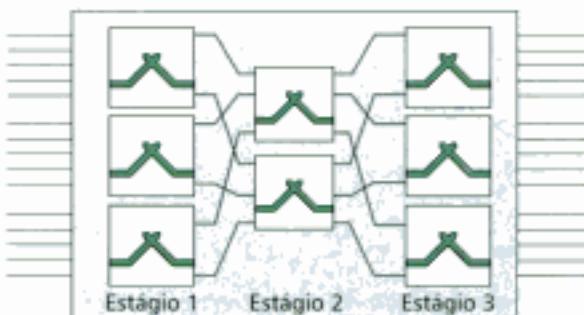


Figura 8.5 Comutador multiestágios.

dores intermediários devem ser conectados a todos os comutadores do primeiro e último estágios de maneira que tenhamos um conjunto de três entradas e três saídas (onde cada conjunto é formado por 5 linhas).

Múltiplos Caminhos ou Rotas Os comutadores multiestágios proporcionam muitas opções de conexão entre cada par de dispositivos nas duas pontas do circuito de comutação. Usando o comutador do exemplo anterior, a Figura 8.6 ilustra duas maneiras de conectar entre si os tráfegos das linhas 4 e 9.

Na Figura 8.6a é estabelecido um caminho entre a linha de entrada número 4 e a linha de saída número 9. Nesta instância, o caminho ou a rota passa pelo comutador intermediário inferior e segue para a primeira entrada do segundo comutador no lado saída, chegando ao destino que é a linha de saída de número 9. A Figura 8.6b ilustra outra rota entre as mesmas linhas de entrada e saída.

Vamos comparar o número de pontos de cruzamento num único comutador matricial 15 por 15 com o número de pontos num comutador multiestágios 15 por 15 descrito anteriormente. Num comutador matricial genuíno precisamos de 255 pontos de cruzamento (15×15). Num comutador multiestágios precisamos:

- Três comutadores no primeiro estágio, cada qual com 10 pontos de cruzamento (5×2), o que resulta num total de 30 pontos para todo o primeiro estágio.
- O segundo estágio é formado por dois comutadores, cada qual com 9 pontos de cruzamento (3×3), o que resulta em 18 pontos para todo o estágio intermediário.
- Três comutadores no estágio final, cada qual com 10 pontos de cruzamento (5×2), o que resulta num total de 30 pontos para todo o terceiro estágio.

O número total de pontos de cruzamento requerido nesse comutador multiestágios vale 78. Nesse exemplo, o comutador multiestágios requer somente 35% de todos os pontos que um comutador matricial necessitaria.

Blocking Esta economia no número de pontos de cruzamento tem um preço. Ela resulta num fenômeno denominado **blocking** durante os períodos de tempo onde o tráfego no circuito de comutação é intenso. Esse fenômeno mede o número de vezes que uma entrada não pode ser conectada a uma saída porque não existe uma rota disponível entre elas, isto é, todas rotas passando pelos comutadores intermediários estão ocupadas.

Num comutador matricial não ocorre o *blocking*. Isso acontece porque cada combinação de entrada e saída possui um ponto específico (único) na matriz, assegurando a existência de uma rota permanente. (Os casos em que duas entradas estão tentando comunicar-se com a mesma saída não devem ser considerados. Nesses casos, a rota não está bloqueada, é a saída que se encontra ocupada.) No comutador multiestágios descrito no exemplo anterior, somente duas entradas, das cinco possíveis, podem ser utilizadas ao mesmo tempo e, consequentemente, somente duas entradas do estágio intermediário podem ser utilizadas simultaneamente e assim por diante. Quanto menor for o número de entradas do estágio intermediário, maiores são as restrições ao número de *links* simultâneos disponíveis num comutador multiestágios.

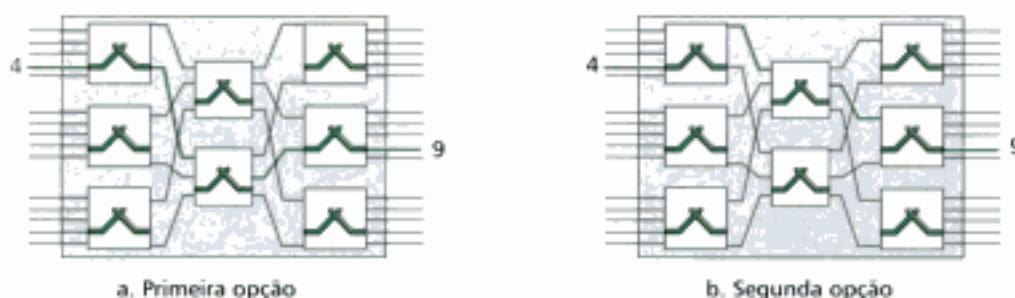


Figura 8.6 Caminhos ou rotas de comutação.

Nos grandes sistemas de comutação, tal como aqueles com mais de 10.000 entradas e saídas, o número de estágios pode ser aumentado de maneira a minimizar o número de pontos de cruzamento exigidos. Entretanto, quando o número de estágios aumenta, a possibilidade de ocorrência de *blockings* aumenta numa proporção bastante parecida. Muitas pessoas já tiveram experiência com o fenômeno *blocking* nos sistemas de telefonia pública, por exemplo, durante algum desastre natural, quando tentavam realizar chamadas para obter notícias de pessoas queridas e o sistema não respondia, pois estava com a carga de ligações excedida. Porém, em circunstâncias normais, o *blocking* não é tratado usualmente como um problema. Em alguns países, as empresas de telefonia não pouparam despesas; elas calculam o número de comutadores entre as linhas de modo a tornar improvável o fenômeno de *blocking*. O método matemático para encontrar o número ideal de comutadores baseia-se em análise estatística, a qual está fora do escopo deste livro.

Comutação por Divisão de Tempo

A comutação por divisão de tempo utiliza a multiplexação TDM para realizar a comutação. Existem dois métodos bastante conhecidos usando a TDM: o *time-slot interchange* e o TDM bus.

Time-Slot Interchange (TSI)

A Figura 8.7 mostra um sistema conectando quatro linhas de entrada às quatro linhas de saída. Imagine que cada linha de entrada deseja enviar dados para uma linha de saída de acordo com o seguinte padrão:

$$1 \rightarrow 3 \quad 2 \rightarrow 4 \quad 3 \rightarrow 1 \quad 4 \rightarrow 2$$

A Figura 8.7a ilustra o resultado da multiplexação TDM ordinária. Como você pode ver, a tarefa pretendida não está sendo realizada. Os dados chegam às saídas na mesma ordem em que eles foram colocados nas entradas. Os dados vão de 1 para 1, 2 para 2 e assim por diante.

Entretanto, na Figura 8.7b, inserimos no link um dispositivo denominado **Time-Slot Interchange (TSI)**. Um TSI troca ou permuta (*interchange*) a ordem dos *slots* baseado nas conexões desejadas. Nesse caso, ele altera a seqüência dos dados de A, B, C, D para C, D, A, B. Sendo assim,

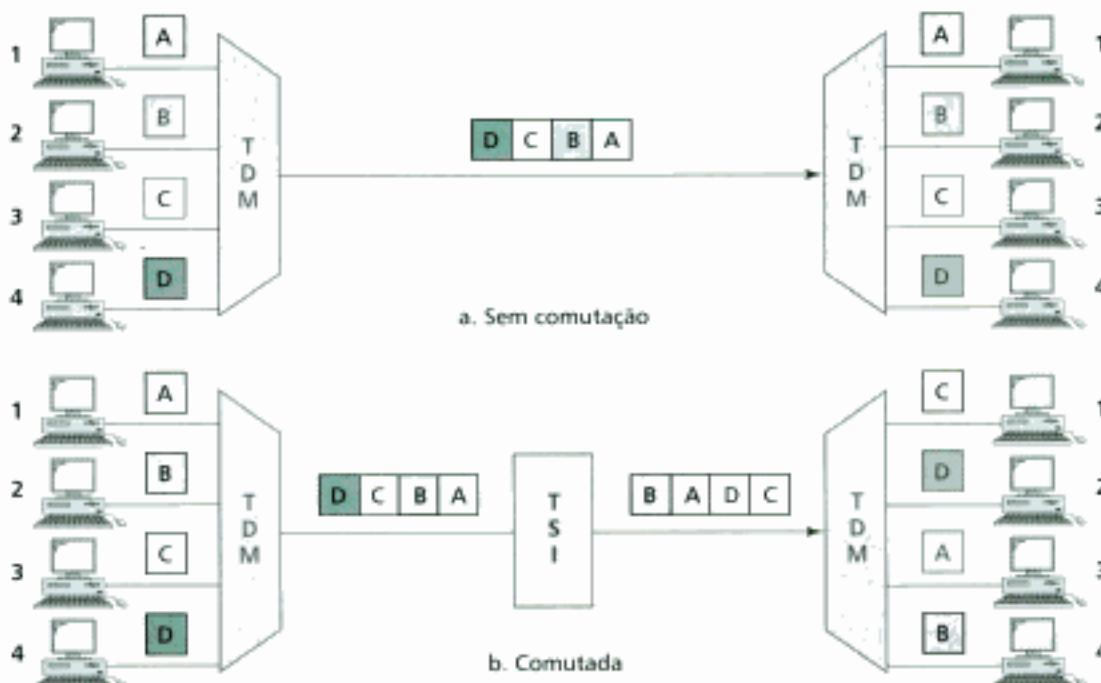


Figura 8.7 TDM com e sem time-slot interchange – TSI.

quando o DEMUX de saída separar os *slots*, ele os passará às saídas corretas, conforme seqüência previamente estabelecida.

A Figura 8.8 mostra como um dispositivo TSI funciona. Um TSI possui uma memória RAM de capacidade relativamente alta. Os endereços na RAM são manipulados de maneira a promover o armazenamento de um *slot* por localização lógica. O número de endereços (localizações) é o mesmo que o número de entradas do comutador (em muitos casos, a quantidade de entradas e saídas também são iguais). A memória RAM é preenchida com os dados de entrada retirados dos *time-slots* na ordem recebida. Em seguida, os *slots* são reenviados para a saída na ordem baseada na unidade de controle.

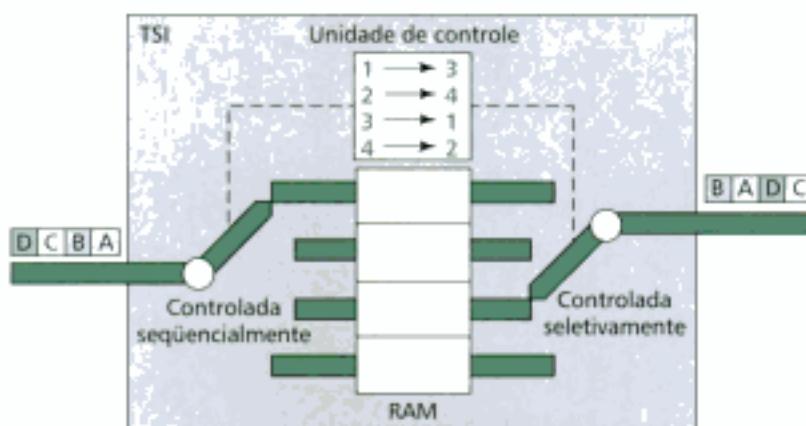


Figura 8.8 Time-slot interchange (TSI).

TDM Bus

A Figura 8.9 é uma versão simplificada de um sistema **TDM bus**. As linhas de entrada e saída são conectadas num barramento (*bus*) de alta velocidade através de portas lógicas de entrada e saída (microchaves). Cada porta de entrada é selecionada durante um dos quatro *time-slots* do *frame* de entrada. Durante esse mesmo *time-slot*, somente uma saída é selecionada. Este par de portas permite uma rajada de dados a serem transferidos da entrada para a saída específica através do barramento. A unidade de controle é a responsável por sincronizar os dispositivos de acordo com a seqüência preestabelecida. Por exemplo, na figura, no primeiro *time-slot*, a porta de entrada 1 e a porta de saída 3 serão selecionadas; durante o segundo *time-slot*, a porta de entrada 2 e a porta de saída 4 serão selecionadas e assim por diante.

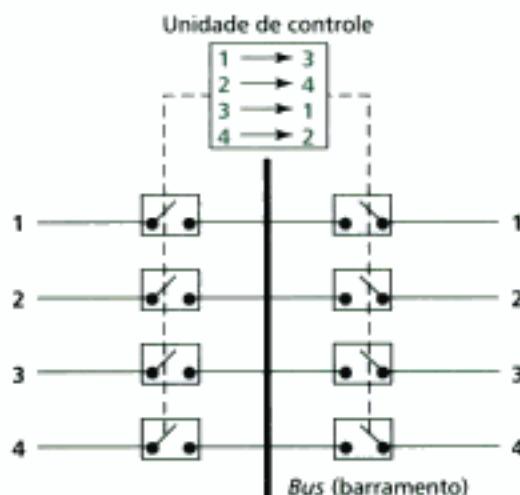


Figura 8.9 TDM bus.

Combinações das Comutações por Divisão de Espaço e de Tempo

Quando comparamos as comutações por divisão de espaço e comutação por divisão de tempo, emergem alguns fatos interessantes. A vantagem da comutação por divisão de espaço é que ela é instantânea. E a desvantagem é o número de pontos de cruzamento necessários na matriz para tornar a comutação por divisão de espaço aceitável em termos de *blockings*.

A vantagem da comutação por divisão de tempo é que ela não necessita de nenhuma matriz com pontos de cruzamento. A desvantagem, no caso da TSI, é que o processamento adicional de cada conexão gera atrasos (*delays*). Primeiramente, cada *time-slot* é armazenado numa seção de memória RAM, então a informação é recuperada e, em seguida, passada adiante.

A terceira opção de comutação é híbrida, ela conjuga as tecnologias por divisão de espaço e por divisão de tempo de modo a tirar as vantagens que cada uma tem. Combinando os dois resultados principais das duas comutações é possível otimizar tanto física (o número de pontos de cruzamento) quanto temporalmente (o valor do *delay*). Comutadores multiestágios deste tipo são projetados para prover as comutações Time-Space-Time (TST), Time-Space-Space-Time (TSST), Space-Time-Time-Space (STTS) e outras combinações possíveis.

A Figura 8.10 mostra um comutador TST simples que consiste em dois estágios temporais e um estágio espacial, disponibilizando ao todo 12 entradas e 12 saídas. Em vez de realizar comutação por divisão de tempo, ele divide as entradas em três grupos de quatro canais cada que as direciona a um TSI de três entradas/saídas. Como resultado, o atraso médio passa a ser um terço do que se esperaria utilizando somente um TSI para prover todas as 12 entradas.

O último estágio é um espelho do primeiro. O estágio intermediário é um comutador por divisão de espaço que conecta grupos TSI para permitir a conectividade entre todos os possíveis pares de entrada/saída (p. ex., conectar a linha de entrada número 3 do primeiro grupo à saída número 7 do segundo grupo).

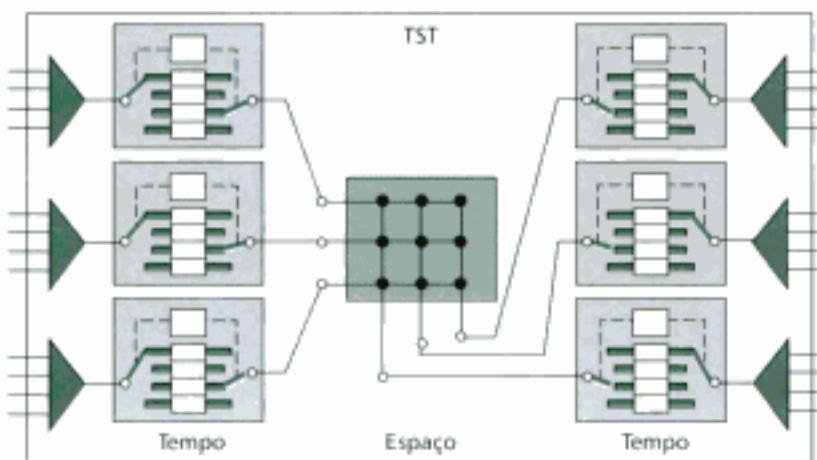


Figura 8.10 Comutador TST.

8.2 REDES DE TELEFONIA

As redes de telefonia fixa usam comutação de circuitos. Elas começaram a ser desenvolvidas no final do século 19. Toda a antiga rede de telefonia, aqui referida como Plain Old Telephone System (POTS), era originalmente um sistema analógico utilizando sinais analógicos para transmitir voz. Com o advento dos computadores, as redes de telefonia passaram a receber dados adicionalmente à voz (início de 1980). Durante a última década, as redes de telefonia experimentaram muitas mudanças tecnológicas. Hoje, a rede é tanto analógica quanto digital.

Componentes Macro de uma Rede

Uma rede de telefonia, como mostra a Figura 8.11, é constituída a partir de três componentes principais: as conexões locais (assinante), os troncos e as centrais de comutação. Ela admite muitos níveis de centrais de comutação, tais como as locais, as interurbanas e as regionais ou interurbanas.

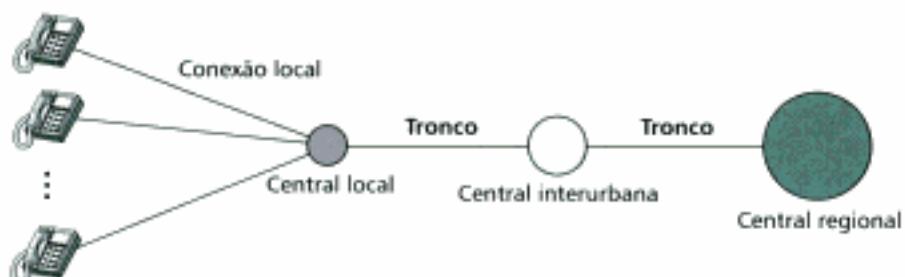


Figura 8.11 Um sistema telefônico.

Linhas de Comunicação Local

O componente da rede telefônica mais próximo ao usuário final é a **linha de comunicação local** que conecta o aparelho telefônico do assinante à **central local** mais próxima. A linha local, quando utilizada para voz, possui uma largura de banda de aproximadamente 4000 Hz (4 kHz). A estrutura do número de cada linha segue um padrão numérico que depende da região. É interessante examinarmos o número do telefone associado a cada conexão local. Os três primeiros dígitos de um número de telefone definem a central local e os outros quatro dígitos definem o número da conexão local dentro da região*.

Troncos

Os **troncos** são os meios de transmissão que desempenham a comunicação entre centrais de comutação. Um tronco disponibiliza normalmente centenas ou milhares de conexões através de multiplexação. Hoje em dia, existe uma predileção em montar troncos de transmissão em fibra óptica ou *links* de satélite.

Central de Comutação

Para evitar conexões físicas permanentes entre dois ou mais assinantes, as companhias telefônicas possuem grandes dispositivos comutadores montados numa **central de comutação**. Uma central desse porte conecta muitos assinantes ou troncos e permite a conexão entre diferentes assinantes em diversas partes do mundo.

LATAs

Após o desmantelamento da AT&T em 1984 (veja a seção *Uma Breve História*), os Estados Unidos foram divididos em mais de 200 **Local Access Transport Areas (LATAs)****. A quantidade de LATAs aumentou desde então. Uma LATA pode ser formada por uma área metropolitana pequena ou grande. Um estado pequeno pode ter uma única LATA; um estado grande pode ter várias LATAs. A fronteira de uma LATA pode sobrepor à fronteira de um estado, isto é, uma parte da LATA pode estar num estado, a outra parte pode estar noutro estado.

* N. de R. T.: No Brasil, muitas cidades ainda dispõem de números de telefones com sete dígitos. Outras tantas já utilizam a nova identificação através de oito dígitos. Sendo assim, os quatro primeiros dígitos definem a central local.

** N. de R. T.: Tanto a história quanto a estrutura do sistema telefônico descritos aqui faz parte da história das comunicações telefônicas nos Estados Unidos. No Brasil, em termos de funcionamento, existe alguma semelhança com o sistema americano.

Serviços Intra-LATA

Os serviços oferecidos pelas **companhias telefônicas** dentro de uma LATA são denominados serviços intra-LATA. A companhia que disponibiliza estes serviços é denominada **Local Exchange Carrier (LEC)**. Antes do ato Telecommunications Act de 1996 (veja a seção *Uma Breve História*), apenas uma companhia tinha concessão dos serviços intra-LATA. As companhias que proviam os serviços antes de 1996 possuíam um sistema de cabeamento próprio e eram denominados **Incumbent Local Exchange Carrier (ILEC)**. As novas companhias que podem prover serviços intra-LATA são denominadas **Competitive Local Exchange Carriers (CLECs)**. Para evitar os custos dos novos sistemas de cabeamento, foi acordado que as ILECs continuariam a prover os serviços principais e as CLECs proveriam outros serviços, tais como os serviços de telefonia móvel, taxas das chamadas dentro de uma LATA e outros. A Figura 8.12 mostra uma LATA e uma central de comutação.

Serviços intra-LATA são providos pelas Local Exchange Carriers (LECs). Desde de 1996, existem dois tipos de LECs: a Incumbent Local Exchange Carriers (ILECs) e a Competitive Local Exchange Carriers (CLECs).

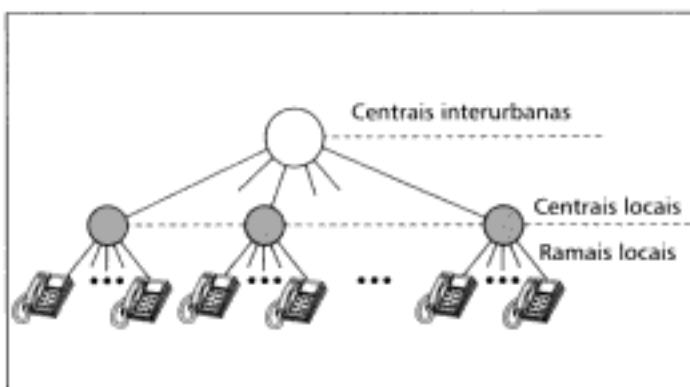


Figura 8.12 Centrais de comutação numa LATA.

As comunicações dentro de uma LATA é realizada pelas centrais locais de comutação e centrais interurbanas. Uma chamada pode ser completada usando somente essas centrais sendo considerada livre de taxas extras. Uma chamada que fizer uso de uma central interurbana (intra-LATA) é sobretaxada.

Serviços Inter-LATA

Os serviços entre LATAs são oferecidos pelas **Interexchange Carriers (IXCs)**. Estas companhias, às vezes denominadas **companhias de longa distância**, provêem serviços de comunicação entre dois consumidores situados em diferentes LATAs. Após o ato de 1996, estes serviços puderam ser providos por qualquer companhia, incluindo aquelas envolvidas apenas com os serviços intra-LATAs. O terreno está aberto. Dentre as maiores companhias que provêem serviços inter-LATAs estão AT&T, MCI, WorldCom, Sprint e Verizon.

As IXCs são companhias de longa distância que provêem serviços de comunicação de dados genéricos, incluindo o serviço de telefonia. Normalmente, uma chamada através de uma IXC é digitalizada, com as companhias usando diversos tipos de redes para prover o serviço.

Pontos de Presença (POPs)

Abordamos os serviços intra-LATAs providos por diversas LECs (uma ILEC e, possivelmente, mais de uma CLEC). Vimos também que os serviços inter-LATAs podem ser providos por muitas IXCs. Daí, surge a questão: como estas companhias interagem umas com as outras? A resposta é: através de um ambiente central de comutação denominado **Ponto de Presença (POP)**. Cada IXC que quiser prover serviços inter-LATAs dentro de uma LATA deve possuir um POP nessa LATA. As LECs que provêem serviços dentro da LATA devem prover conexões de modo que cada assinante tenha acesso a todos os POPs. A Figura 8.13 ilustra o conceito.

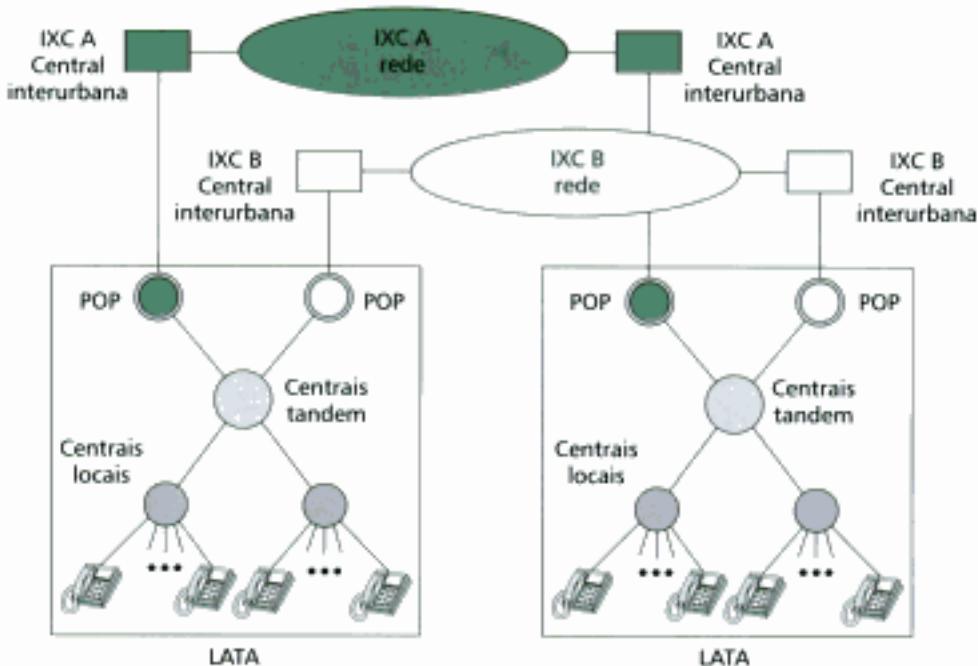


Figura 8.13 POPs.

Um assinante que precisar fazer uma chamada até outro assinante distante é conectado primeiramente, através da linha de comunicação local, a uma central de comutação local para então ser conduzido até uma central comutadora interurbana e depois chegar ao nível POP, ou então, ser conectado diretamente a um POP. Nesse caso, as chamadas partem do POP de uma IXC (onde o assinante foi escolhido) na LATA fonte até o POP da mesma IXC na LATA destino. A chamada passará através de centrais interurbanas da IXC e será encaminhada através da rede provida pela IXC.

Fazendo uma Chamada

As linhas telefônicas dos assinantes são conectadas através das conexões locais às centrais de comutação locais.

Para acessar uma estação de comutação dentro da central é necessário realizar uma discagem. No passado, praticamente todos os aparelhos telefônicos apresentavam um disco de discagem de onde as ligações telefônicas partiam na forma de pulsos. Para cada número escolhido e discado era associado um sinal digital prontamente enviado à central mais próxima. Este tipo de discagem ficava sujeita a muitos erros de manuseio devido à inconsistência humana durante o processo de discagem.

Hoje em dia, o disco foi substituído por teclas e os pulsos de discagem cada vez mais estão sendo substituídos por tons. Nesse método, em vez de enviar um sinal digital, o usuário envia duas pequenas "rajadas" ou sequências de sinais analógicos de freqüências diferentes, chamada *tom dual*. A freqüência dos sinais analógicos enviados depende da linha e da coluna onde a tecla está posicionada.

A Figura 8.14 ilustra um sistema a disco e outro de teclas. Nessa figura, quando um usuário discar, por exemplo, o número 8 um sinal digital que o representa será gerado na linha. De outro modo, se um usuário pressionar a tecla referente a esse mesmo número, um sinal analógico em rajada de duas freqüências moduladas (852 e 1336Hz) é enviado à linha telefônica.

No passado, a comunicação de voz utilizava sinais analógicos. Hoje em dia usa sinais digitais. Na contramão da evolução dos sinais, antes a discagem enviava sinais digitais na linha, representando números registrados em um disco. Hoje, o processo de realização de uma chamada envia sinais analógicos na linha e utiliza para isso um teclado.

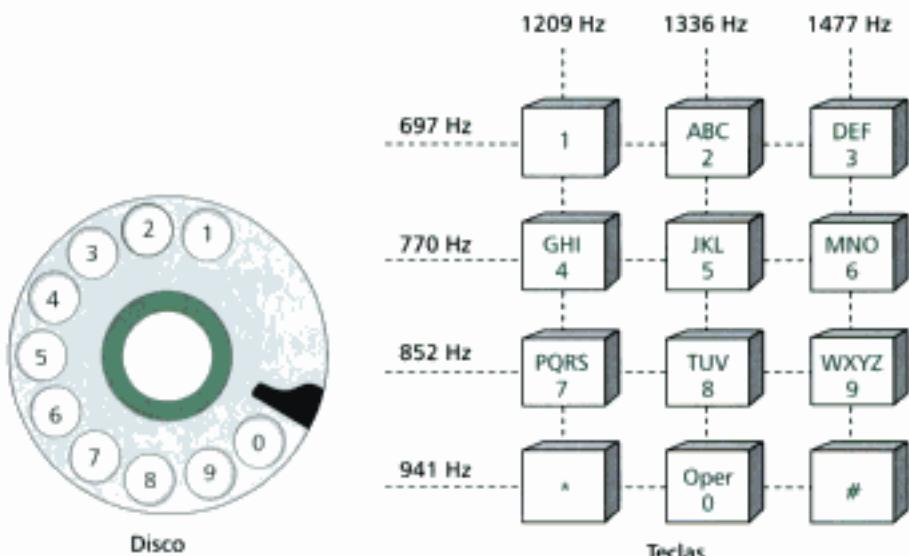


Figura 8.14 Discagem por pulso versus discagem por tom.

Serviços Analógicos

Nas primeiras décadas da era telefônica, as companhias proviam aos usuários serviços de telefonia analógicos. Estes serviços continuam até nos dias de hoje. Classificamos estes serviços como serviços de comutação analógica.

Serviço de Comutação Analógica

O **serviço de comutação analógica** é o tipo de serviço mais comum que uma companhia provê aos usuários domésticos. O sinal na linha de comunicação local entre o usuário e a central é analógico e a largura de banda fica geralmente entre 0 e 4000 Hz. Utilizando linhas comutadas, quando a pessoa que realiza a chamada disca o número desejado, a chamada é transportada a um comutador ou uma série de comutadores para o estabelecimento da conexão. Os comutadores apropriados são então ativados no *link* da linha da pessoa desejada. O(s) comutador(es) conecta(m) as duas linhas enquanto durar a chamada.

As LECs e IXCs provêm serviços opcionais aos consumidores residencial e de empresa. Vejamos os mais comuns.

Serviços de Chamadas Locais Um serviço de **chamada local** normalmente é contratado por taxas fixas mensais, embora, em algumas LATAs, as companhias telefônicas taxem cada chamada ou conjunto de chamadas*. Os planos pagos por chamada são recomendados às pessoas que normalmente não têm o hábito de realizar muitas chamadas.

Serviços de Chamadas Interurbanas Uma **chamada interurbana** pode ficar restrita à intra-LATA ou inter-LATA. Se a LATA for extensa geograficamente, uma chamada pode passar através de muitas centrais interurbanas e o custo da ligação pode ser elevado para o assinante. As chamadas de longa distância (inter-LATAs) costumam ser salgadas por esse motivo.

Serviços 800 Se um assinante (normalmente uma empresa ou organização) precisar prover chamadas gratuitas para outros assinantes (normalmente os consumidores), ele pode solicitar, junto a alguma companhia telefônica, um **serviço 800** (esse número pode ser 888, 877 ou 866 porque os números 800 estão cada vez menos disponíveis). Neste caso, a chamada é grátis para quem faz a ligação, mas é paga pela empresa que contratou o serviço. O serviço pode ficar restrito a um único estado, a um grupo de estados ou a todo um país. As taxas cobradas serão proporcionais à extensão territorial de utilização do número.

* N. de R. T.: Os serviços e números comentados aqui são padronizados nos Estados Unidos. Alguns também podem ser encontrados no Brasil.

WATS As **Wide-Area Telephone Services (WATS)** são os serviços opostos ao 800/888. Os serviços 800 são chamadas recebidas por uma organização que disponibiliza algum serviço de utilidade pública. As WATS são taxadas através do número de chamadas realizadas por uma organização ou empresa. Este serviço pode ser menos oneroso que o serviço tradicional de chamadas interurbanas, porque são cobrados de acordo com a quantidade de ligações. Este serviço está disponível nos níveis estadual, regional ou nacional. Outra vez, os contratos levam em conta a extensão territorial das chamadas.

Serviços 900 Os **serviços 900** são parecidos com os serviços 800/888, que são chamadas recebidas de um grupo de assinantes. Entretanto, diferentemente dos serviços 800/888, a chamada é paga pela pessoa que liga e normalmente é mais cara que uma ligação de longa distância. O motivo dessa taxa ser mais elevada é que uma companhia normalmente faz duas cobranças: a primeira pela chamada interurbana em si e a segunda pelos serviços pagos à empresa ou organização que mantém a linha. Este serviço normalmente é utilizado por organizações que taxam os consumidores ou clientes pelos serviços prestados. Por exemplo, uma companhia de *software* pode precisar cobrar dos clientes pelo suporte técnico.

Serviços de Aluguel de Linhas Analógicas

Um **serviço de aluguel de linhas analógicas** oferece aos consumidores a oportunidade de alugar uma linha, chamada às vezes de *linha dedicada*, e conectá-la permanentemente a outro usuário. Embora essa conexão ainda passe por centrais de comutação numa rede telefônica, os assinantes tratam-na como uma linha única porque o circuito de comutação está sempre disponível, dispensando até mesmo o processo de discagem.

Serviços Digitais

Recentemente, as companhias telefônicas começaram a oferecer **serviços digitais** aos assinantes. Os serviços digitais são menos sensíveis aos ruídos e outras formas de interferência que os serviços analógicos. Os dois tipos de serviços digitais mais comuns são o *switched/56* e o serviço ou rede digital de dados (DDS)*. Já examinamos um tipo de serviço digital de alta velocidade: as linhas T (americana) e E (européia) no Capítulo 6. Abordaremos o acesso de alta velocidade residencial no Capítulo 9.

Serviço Switched/56

O serviço **switched/56** é a versão digital de uma linha analógica comutada. Ele é um serviço de comutação digital que permite taxas de transmissão de até 56 kbps. Para se comunicar através deste serviço, ambas as partes devem assiná-lo. A pessoa que faz a chamada utilizando um serviço telefônico comum não pode se conectar a um computador ou telefone que utilize um *switched/56*, até mesmo se um modem for utilizado. De outro modo, os serviços analógico e digital representam domínios diferentes para as companhias telefônicas.

Como a linha é totalmente digital num serviço *switched/56*, os assinantes não necessitam de modems para transmitir dados digitais. Entretanto, eles precisam de outro dispositivo denominado **Digital Service Unit (DSU)**. Tal dispositivo converte a taxa de dados gerados pelo computador do assinante para 56 kbps, codificando-os num formato inteligível ao provedor de serviços. O *switched/56* suporta banda sobre demanda permitindo que os usuários obtenham altas velocidades utilizando mais de uma linha (veja a seção sobre multiplexação inversa no Capítulo 6). Esta opção faz com que o serviço *switched/56* suporte vídeo conferência, *fast fax*, multimídia, altas taxas de dados, dentre outros serviços.

Serviço ou Rede Digital de Dados (DDS)

Uma **DDS** é a versão digital da linha dedicada (alugada). Ela é uma linha digital que suporta velocidades de transmissão de até 64 kbps**.

* N. de R. T.: No Brasil, os serviços digitais são encontrados nas Redes Digitais de Serviços Integrados (RDSI).

** N. de R. T.: Na época em que o livro foi publicado. Hoje em dia, podemos contratar serviços de dados de 512 kbps a preços atraentes.

Uma Breve História

Antes de fechar o capítulo, vamos rever a história das companhias telefônicas nos Estados Unidos. Esta história pode ser dividida em três fases: antes de 1984; entre 1984 e 1996 e após 1996.

Antes de 1984

Antes de 1984, quase toda a comunicação de curta e longa distância era provida pela AT&T Bell System. Em 1970, o governo americano acreditava que a Bell System estava monopolizando os serviços de telefonia, então processou a companhia. O veredito foi favorável ao governo e, baseado num documento cujo nome era Modified Final Judgment (MFJ), em 01/01/1984 o governo desmantelou a AT&T em AT&T Long Lines, 23 Bell Operating Companies (BOCs) e outras. As 23 BOCs foram agrupadas juntas de modo a construir muitas Regional Bell Operating System (RBOCs). Esta decisão extraordinária, a divisão da AT&T em 1984, foi muito benéfica ao usuários dos serviços telefônicos. Os valores cobrados pelas empresas telefônicas caíram vertiginosamente.

Entre 1984 e 1996

O desmantelamento da AT&T dividiu os Estados Unidos em mais de 200 LATAs; algumas companhias tiveram permissão de prover serviços apenas dentro de uma LATA (LECs), outras foram permitidas operar entre LATAs (IXCs). A competição, particularmente entre as empresas de longa distância, impulsionou o surgimento de novas companhias. Entretanto, nenhuma LEC tinha permissão de operar serviços de longa distância e nem muito menos as IXC poderiam operar serviços locais.

Após 1996

Outra mudança gigantesca nas telecomunicações nos Estados Unidos ocorreu em 1996. O ato denominado Telecommunications ACT de 1996 combinou os diferentes serviços providos pelas diferentes empresas de telefonia num único guarda-chuva de serviços de telecomunicações; incluindo serviços locais, serviços de voz em longas distâncias e serviços de dados, vídeo, dentre outros. Além disso, o ato liberou qualquer companhia a prover qualquer um desses serviços nas esferas local e de longa distância. Noutras palavras, uma mesma companhia recebeu permissão para operar tanto dentro de LATAs quanto entre LATAs. Contudo, para evitar que novos sistemas de cabeamento fossem lançados, a companhia que operava serviços intra-LATAs (ILEC) continuou a prover os serviços principais. Aos novos competidores coube ofertar outros serviços.

8.3 TERMOS-CHAVE

<i>Blocking</i>	Local Access and Transport Area (LATA)
Central de comutação	Local Exchange Carrier (LEC)
Central local	Nó de comutação
Companhia de longa distância	Ponto de cruzamento
Companhias telefônicas	Ponto de Presença (POP)
Competitive Local Exchange Carrier (CLEC)	Serviço 800
Comutação	Serviço 900
Comutação de circuitos	Serviço de aluguel de linhas
Comutação multiestágios	Serviço de chamada interurbana
Comutação por divisão de tempo	Serviço de comutação analógica
Comutação por divisão do espaço	Serviço digital
Conexão ponto a ponto	Serviço Digital de Dados (DDS)
Digital Service Unit (DSU)	Serviço switched/56
Discagem por pulso	Serviços de chamada local
Discagem por tom	TDM bus
Incumbent Local Exchange Carrier (ILEC)	Time-Slot Interchange (TSI)
Interexchange Carrier (IXC)	Tronco
Linha de comunicação local	Wide-Area Telephone Service (WATS)

8.4 RESUMO

- Comutação é um método para conexão eficiente de muitos dispositivos de comunicação.
- Um nó de comutação é um recurso de *hardware* ou *software* que interliga temporariamente dispositivos de comunicação.
- Existem três tipos fundamentais de comutação: de circuitos, de pacotes e de mensagens.
- Na comutação de circuitos, é estabelecida uma conexão física direta entre dois dispositivos através da comutação por divisão de espaço, comutação por divisão de tempo ou ambas.
- Numa comutação por divisão de espaço, um caminho ou rota interligando dois dispositivos é separado espacialmente dos demais.
- Um comutador matricial é o dispositivo de comutação mais comum na comutação por divisão de espaço. Ele conecta n entradas a m saídas formando $n \times m$ pontos de cruzamento.
- Comutadores multiestágios podem reduzir o número de pontos de cruzamento, mas resultam muitas vezes em *blockings*.
- Os *blockings* ocorrem quando nem todas as entradas possuem um caminho correspondente único até a saída.
- Numa comutação por divisão de tempo, as entradas são separadas no tempo usando a TDM. Uma unidade de controle envia a entrada para o dispositivo de saída correto.
- O Time-Slot Interchange (TSI) e o TDM bus são dois tipos de comutação por divisão de tempo.
- As comutações por divisão de espaço e de tempo podem ser combinadas.
- Uma rede de telefonia pública é um exemplo de rede de comutação de circuitos.
- Um sistema de telefonia fixa é formado por três grandes componentes, a saber: a conexão local, os troncos e as centrais de comutação.
- Os Estados Unidos foram divididos em mais de 200 Local Access and Transport Areas (LATAs).
- Serviços intra-LATAs são providos pelas Incumbent Local Exchange Carriers (ILECs) e as Competitive Local Exchange Carriers (CLECs). Os serviços inter-LATAs são oferecidos pelas Interchange Carriers (IXCs).
- As companhias telefônicas provêem serviços de comutação analógica do tipo chamada local e interurbana, serviços 800/888, WATS e serviços 900.
- As empresas de telefonia provêem dois serviços de comutação digital: o *switched/56* e a rede digital de dados.
- O monopólio da AT&T foi quebrado, em 1984, através de um processo movido pelo governo americano.

8.5 PRATIQUE OS CONHECIMENTOS ADQUIRIDOS

Questões de Revisão

1. Quais são os três maiores métodos de comutação?
2. Cite os dois métodos de comutação de circuitos existentes.
3. O que é um ponto de cruzamento num comutador matricial?
4. Que fator limita o uso de um comutador matricial? Um comutador multiestágios resolve definitivamente o problema?
5. Como o processo de *blocking* se relaciona com o comutador matricial?
6. Como o processo de *blocking* se relaciona com o comutador multiestágios?
7. Compare os mecanismos de comutação por divisão de espaço e por divisão de tempo.
8. Cite as duas tecnologias utilizadas na comutação por divisão de tempo.
9. Compare a técnica TSI com a TDM *bus*.
10. Qual é a função de uma unidade de controle num TSI e num TDM *bus*?
11. Qual a vantagem da comutação por divisão de espaço sobre a comutação por divisão de tempo?
12. Qual a vantagem da comutação por divisão de tempo sobre a comutação por divisão de espaço?
13. Quais são os três grandes componentes de um sistema de telefonia?
14. O que é uma conexão local?
15. Qual é largura de banda de uma linha telefônica tradicional?
16. Qual é a função de um tronco?

17. Como uma ILEC é diferente de uma CLEC?
18. Qual é a função de um POP?
19. Como são realizados os serviços telefônicos entre LATAs?
20. Compare os sinais utilizados num sistema discado com os sinais num sistema teclado.
21. De que forma o serviço 800 difere do serviço 900?
22. Qual é a diferença entre um serviço de comutação analógico e um serviço de aluguel de linha?
23. Qual é a função de uma DSU?

Questões de Múltipla Escolha

24. O _____ é um dispositivo que conecta n entradas a m saídas.
 - Ponto de cruzamento
 - Comutador matricial
 - Modem
 - RAM
25. Quantos pontos de cruzamentos existem num comutador matricial de 40 entradas para 50 saídas?
 - 40
 - 50
 - 90
 - 2000
26. Num comutador matricial com 1000 pontos de cruzamento, aproximadamente quantos pontos podem ser utilizados simultaneamente?
 - 100
 - 250
 - 500
 - 1000
27. A _____ de um TSI controla a ordem de entrega dos dados nos *slots*, antes armazenados em RAM.
 - Função do comutador matricial
 - Função do ponto de cruzamento
 - Unidade de controle
 - A função do *transceiver*
28. Numa comutação _____, a entrega de dados é retardada porque os dados devem ser armazenados em RAM e retransmitidos em seguida.
 - Por divisão de espaço
 - Por divisão de tempo
 - Virtual
 - De pacotes
29. Para criar um _____, combinamos vários comutadores matriciais formando estágios.
 - Comutador multiestágios
 - Ponto de cruzamento
 - Comutador de pacotes
 - TSI
30. Qual das seguintes opções é uma comutação por divisão de tempo?
 - TSI
 - TDM bus
 - Ponto de cruzamento
 - (a) e (b)
31. Numa comutação por divisão de tempo, uma _____ governa a entrega de pacotes armazenados em RAM.
 - TDM bus
 - Crosspoint*
 - Crossbar*
 - Unidade de controle
32. Um rede de telefonia pública é um exemplo de rede de _____.
 - Comutação de pacotes
 - Comutação de circuitos
 - Comutação de mensagens
 - Nenhuma das opções anteriores
33. As conexões locais possuem cabos _____ que conectam um assinante a uma central de comutação mais próxima.
 - Par trançado
 - Coaxiais
 - Fibra óptica
 - (a) e (c)
34. Os troncos são meios de transmissão que utilizam _____ para realizar a comunicação telefônica entre centrais.
 - Cabo par trançado
 - Cabo de fibra óptica
 - Links* de satélite
 - (b) e (c)
35. Antes de 1996, uma companhia telefônica que provia serviços numa LATA e todo o sistema de cabeamento dela era denominada _____.
 - ILEC
 - CLEC
 - IXC
 - POP

36. Após 1996, uma nova companhia telefônica operando dentro de uma LATA passou a ser denominada _____.
 a. ILEC
 b. CLEC
 c. IXC
 d. POP
37. Um serviço telefônico realizado entre duas LATAs é denominado _____.
 a. ILEC
 b. CLEC
 c. IXC
 d. POP
38. Se uma central local recebe duas raias de sinais analógicos de freqüências 697 e 1477 Hz, ela irá interpretá-los como o número _____.
 a. 1
39. Dados dentro de um computador são _____; a linha de comunicação local transmite sinais _____.
 a. Analógicos; analógicos
 b. Analógicos; digitais
 c. Digitais; digitais
 d. Digitais; analógicos
40. Uma rede de telefonia pública possui uma banda de aproximadamente _____.
 a. 2000Hz
 b. 4000Hz
 c. 2000MHz
 d. 4000MHz

Exercícios

41. Quantos pontos de cruzamento serão necessários se desejarmos utilizar um comutador matricial para conectar 1000 telefones numa cidade pequena?
42. Determine o número de pontos de cruzamento no esquema da Figura 8.15.
43. Quantos pontos de cruzamento serão necessários se estivermos utilizando somente um comutador matricial, com o mesmo número de entrada e de saída, em vez do esquema mostrado da Figura 8.15?
44. Usando os exercícios 42 e 43, quanto a eficiência do esquema é melhorada se utilizarmos três estágios em vez de um?
45. Na Figura 8.15, quantos usuários conectados em cada comutador do primeiro estágio podem acessar o sistema ao mesmo tempo? Quantos usuários po-

dem acessar o sistema como um todo? Existe uma relação entre a primeira e a segunda respostas desse exercício? Poderíamos afirmar que a segunda resposta pode ser obtida a partir da primeira?

46. Na Figura 8.15, poderíamos aliviar os problemas de *blocking* adicionando mais um estágio intermediário?
47. Qual dos esquemas de três estágios mostrados na Figura 8.16 tem a melhor *performance* em termos de *blocking*? Justifique sua resposta. Determine o número de conexões de entrada/saída para os sistemas intermediários.
48. Qual é a fórmula para determinar, num comutador de três estágios, o número de pontos de cruzamento (n) em termos do número de linhas entrada/saída (N), do número de primeiro e terceiro estágios formado por K comutado-

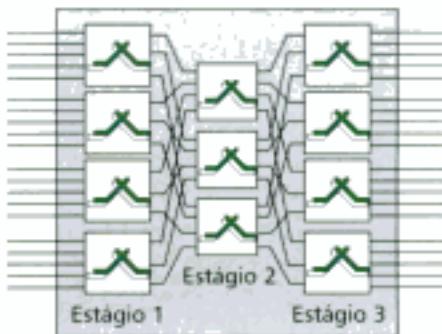


Figura 8.15 Exercícios 42, 43, 45 e 46.

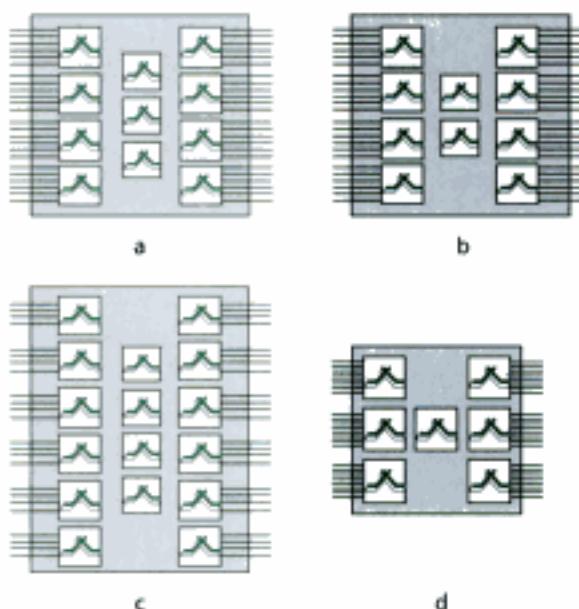


Figura 8.16 Exercício 47.

res e do número de estágios intermediários contendo L comutadores?

49. Na Figura 8.9, quais são as linhas de saída se as linhas de entrada receberem A, B, C e D?
50. Desenhe um esquema TDM *bus* de quatro linhas.

51. Implemente um comutador TSSST com 48 entradas e 48 saídas. Utilize multiplexadores de entrada 4×1 e demultiplexadores de saída 1×4 .

52. Implemente um comutador STS com 10 entradas e 10 saídas. Utilize comutadores 5×2 no primeiro estágio e comutadores 2×5 no último.

Acesso Digital de Alta Velocidade: DSL, Cable Modems e SONET

Vimos no Capítulo 5 como os modems são capazes de criar um sinal digital e como eles possibilitam aos usuários de PC acessarem a Internet através da rede de telefonia pública.

Além disso, vimos que os modems tradicionais impõem um limite superior teórico para taxa de transmissão de dados nesse tipo de acesso. Neste capítulo, abordaremos três tecnologias dominantes que superam os limites dos modems tradicionais: a tecnologia DSL, *cable modem* e SONET.

9.1 TECNOLOGIA DSL

Quando os modems normais atingiram o limite de capacidade de transmissão de dados da Rede de telefonia pública comutada (RTPC), as empresas de telefonia foram compelidas a desenvolver outras tecnologias que permitissem acessar a Internet em altas velocidades. A tecnologia **Digital Subscriber Line (DSL)**, linha digital do assinante, é uma das mais promissoras em suportar comunicação digital em altas velocidades. De fato, a tecnologia DSL forma um conjunto de tecnologias, identificadas por uma letra prefixo (ADSL, VDSL, HDSL e SDSL). Freqüentemente, nos referimos ao conjunto como *xDSL*, onde *x* pode ser substituído por A, V, H ou S.

ADSL

A primeira tecnologia do conjunto foi a **Asymmetrical DSL (ADSL)**. Assim como o modem de 56k, a tecnologia ADSL provê altas taxas de transmissão maiores (em bps) na direção de descida ou em *downstream* (da Internet para o computador do usuário) do que na direção de subida ou *upstream* (do computador do usuário para a Internet). Esta é a razão dessa tecnologia ser denominada assimétrica. Contudo, diferentemente da assimetria dos modems de 56k, as especificações das implementações ADSL dividem de maneira desigual a largura de banda do canal local do usuário. Este serviço não está disponível aos consumidores não residenciais que necessitam de uma banda larga nas duas direções (por exemplo, para empresas).

**ADSL é uma tecnologia de comunicação assimétrica desenvolvida para usuários residenciais.
não está disponível para empresas.**

Usando o Canal Local de Comunicação

Um ponto interessante a favor da tecnologia ADSL é que ela utiliza a mesma conexão local do usuário com a rede telefônica. Então, como a ADSL supera as taxas de transmissão dos modems tradicionais? A resposta está na conexão local que utiliza pares trançados cuja banda de transmissão atinge até 1,1 MHz. Um filtro instalado na casa do usuário seleciona a banda do canal de voz (4 kHz) para as comunicações de voz e uma outra banda para o canal ADSL. Isso é feito de modo a permitir a multiplexação de um grande número de canais de voz. Se o filtro for removido, toda a banda fica disponível de volta para comunicação de dados e voz.

A conexão local de um usuário possui uma largura de banda de 1,1 MHz.

Tecnologia Adaptativa

Infelizmente, a banda de 1,1 MHz para o canal local existe apenas em teoria. Fatores tais como a distância entre a residência e a central de comutação, o tamanho do cabo e o tipo de sinalização de linha utilizada afetam a largura de banda do meio. Os projetistas da tecnologia ADSL conhecem esses problemas e, por isso, desenvolveram uma tecnologia adaptativa que testa as condições e a banda disponível da linha antes de iniciar o processo de transmissão de dados. Assim, a taxa de transmissão de dados da ADSL não é fixa; ela varia de acordo com a condição e o tipo de conexão local.

ADSL é uma tecnologia adaptativa. A taxa de transmissão de dados do sistema depende das condições da linha local.

DMT

A técnica de modulação padronizada para a ADSL é denominada **Discrete Multitone Technique (DMT)** a qual combina as modulações QAM e FDM. Não existe uma regra rígida de divisão da banda de um sistema. De um modo peculiar, cada sistema pode realizar a divisão da banda disponível. Tipicamente, uma banda de 1,104 MHz é dividida em 256 canais. Cada canal utiliza uma largura de banda de 4,312 kHz, de acordo com a Figura 9.1.

A Figura 9.2 mostra como a largura de banda total é dividida nos seguintes canais:

- **Voz.** O canal 0 é reservado para a comunicação de voz.
- **Reserva.** Os canais 1 a 5 são utilizados como banda reservada (banda de proteção) entre as comunicações de voz e de dados.
- **Controle e transmissão de dados em upstream.** É formado a partir dos canais de 6 a 30 (25 canais) utilizados para a transmissão e controle de dados em *upstream*. Um dos ca-

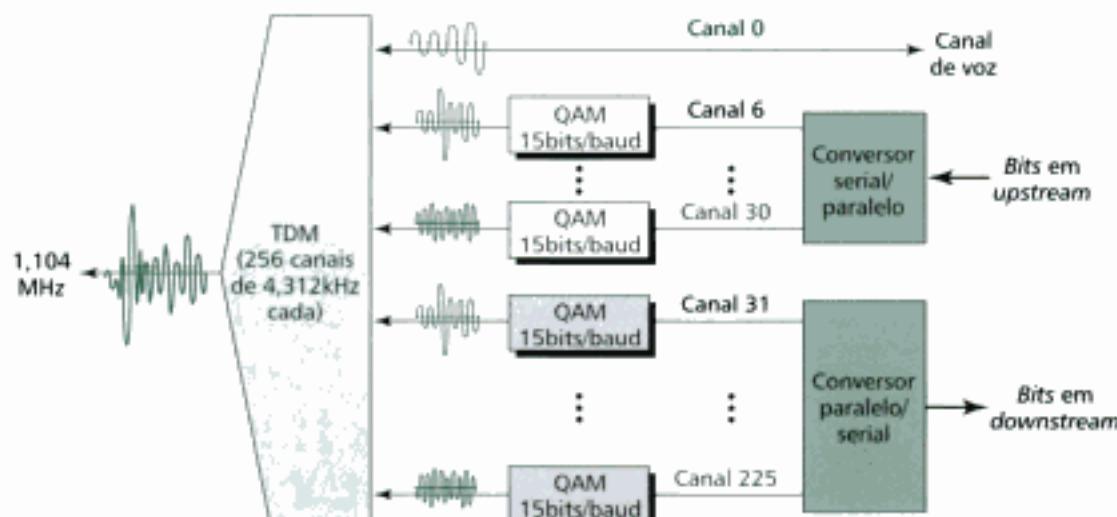


Figura 9.1 DMT.



Figura 9.2 Divisão da banda.

nais é utilizado para controle e os outros 24 são para tráfego de dados. Como são 24 canais, cada um utilizando 4 kHz de banda (diferente dos 4.312 kHz disponíveis no canal 0) e modulação QAM, temos $24 \times 4000 \times 15$, resulta numa taxa de 1,44 Mbps na direção de subida (*upstream*).

- **Controle e transmissão de dados em *downstream*.** É formado a partir dos canais de 31 a 255 (225 canais) utilizados para a transmissão e controle de dados em *downstream*. Outra vez, um canal é utilizado para controle enquanto que os outros 224 são para tráfego de dados. Como são 224 canais, chegamos a uma capacidade final de $224 \times 400 \times 15$ ou 13,4 Mbps.

Taxa de Transmissão Real

Devido ao alto nível de ruído nas linhas telefônicas, a taxa de dados real é muito inferior aos valores supracitados. As taxas reais normalmente se aproximam do seguinte:

Upstream: 64 kbps a 1 Mbps

Downstream: 500 kbps a 8 Mbps

Usuário: Modem ADSL

A Figura 9.3 ilustra um modem ADSL instalado na residência de um usuário de Internet. A conexão local é ligada ao filtro que separa o canal de voz dos canais de dados. O modem ADSL modula os dados (usando DMT) e gera os canais *upstream* e *downstream*.

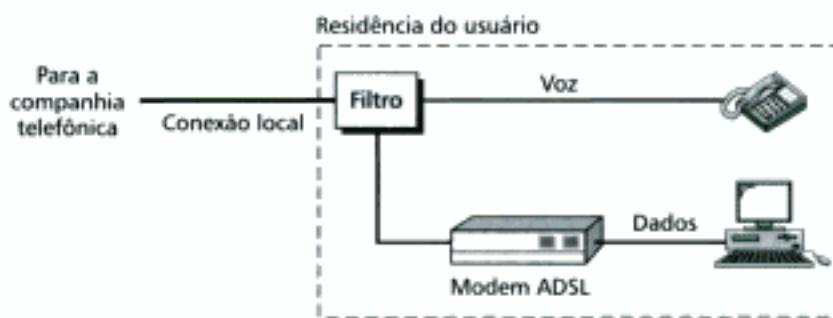


Figura 9.3 Modem ADSL.

Companhia Telefônica: DSLAM

A situação é diferente do ponto de vista da companhia telefônica. Ao invés de um modem ADSL, é instalado um dispositivo denominado **multiplexador de acesso DSL (Digital Subscriber Line Access Multiplexer – DSLAM)** cuja funções são similares ao ADSL. Além do mais, o DSLAM empacota os dados a serem enviados para a Internet (provedor ISP). A Figura 9.4 ilustra a configuração.

Outras Tecnologias DSL

SDSL

ADSL provê comunicação assimétrica. A taxa em *downstream* é muito maior que a taxa em *upstream*. Embora esta característica atenda as expectativas da maioria dos usuários residenciais, ela não é apro-

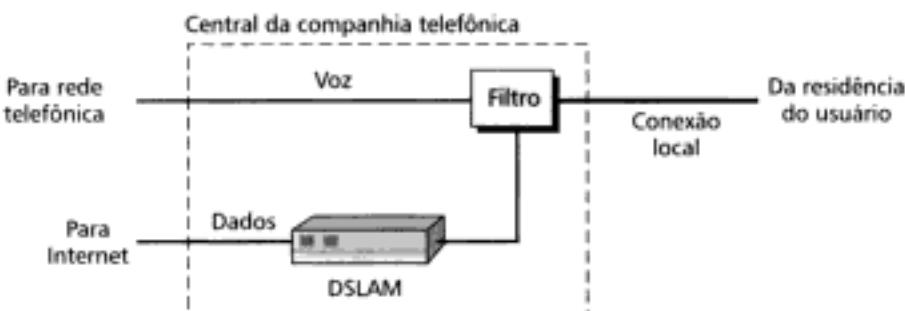


Figura 9.4 DSLAM.

priada para empresas, que geralmente enviam e recebem massas de dados em ambas as direções. A tecnologia **Symmetric Digital Subscriber Line (SDSL)** foi desenvolvida para atender essas expectativas das empresas. Nela, uma banda é dividida igualmente entre as direções de *upstream* e *downstream*.

HDSL

A tecnologia **High-bit-rate Digital Subscriber Line (HDSL)** foi desenvolvida com uma alternativa para a linha T-1 (1,544 Mbps). A linha T-1 utiliza codificação AMI (*Alternate Mark Inversion*) o que a torna muito susceptível à atenuação em altas freqüências. Isto limita a extensão de uma linha T-1 a 1 km. Para longas distâncias é necessário instalar repetidores, o que obviamente eleva os custos da transmissão.

A tecnologia HDSL usa codificação 2B1Q (veja Capítulo 4) que é bem menos suscetível à atenuação. Assim, taxas de dados da ordem de 2 Mbps podem ser obtidas, sem o uso de repetidores, em distâncias de até 3,6 km. O canal da conexão HDSL usa dois pares trançados para implementar o modo de transmissão *full-duplex*.

VDSL

A variante **Very-high-bit-rate Digital Subscriber Line (VDSL)** é uma alternativa bastante similar a ADSL. Entretanto, admite cabo coaxial, fibra óptica ou par trançado como meio de transmissão para comunicação a curtas distâncias (300 a 1800 m). A técnica de modulação é a DMT a qual permite taxas de dados de 50 a 55 Mbps em *downstream* e de 1,5 a 2,5 Mbps em *upstream*.

9.2 CABLE MODEM

Atualmente, as operadoras de TV a cabo estão competindo com as companhias telefônicas pelos usuários residenciais que cada vez mais pedem banda para acessar a Internet em altas velocidades. Como vimos, a tecnologia DSL permite acessos de alta velocidade aos usuários através da conexão telefônica local. Contudo, a tecnologia DSL utiliza pares trançados sem blindagem e, por isso, são muito suscetível a interferências. Isto reflete na conexão, modificando o limite superior da taxa de transmissão. Outra proposta é utilizar a rede de TV a cabo para acessar a Internet. Nesta seção, faremos uma breve introdução à tecnologia Cable Modem.

Redes CATV

Em meados de 1940, a **TV a cabo** entrou em funcionamento para distribuir sinais de *broadcast* de vídeo para as localidades onde a recepção da TV aberta não existia ou era ruim. Ela foi chamada de **Community Antenna TV (CATV)** porque uma antena instalada bem no alto de uma colina ou construção mais elevada recebia os sinais das estações de TV e os distribuía à comunidade via cabos coaxiais. A Figura 9.5 mostra um diagrama esquemático da rede de TV a cabo.

O centro de distribuição da TV a cabo, conhecido como **cabeça de rede (head end)**, recebia os sinais de vídeo das estações de *broadcasting* e os utilizava para alimentar a rede de cabos coaxiais que, por sua vez, conduzia os sinais até as casas. Como a tecnologia da época era precária, era necessário instalar muitos amplificadores entre a cabeça de rede e as residências, porque a in-

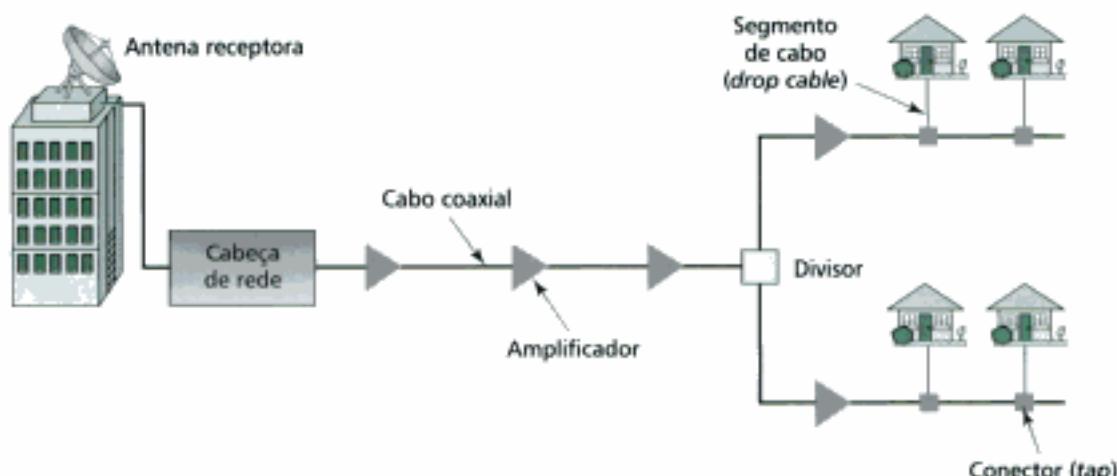


Figura 9.5 Rede de TV a cabo convencional.

tensidade do sinal enfraquecia à medida que o sinal viajava pelo cabo. A rede admitia a instalação de até 35 amplificadores entre a cabeça de rede e o assinante final. No outro extremo, próximo às residências, eram instalados divisores de sinal, conectores de derivação (*tags*) e pequenos segmentos de cabos que levavam o sinal do cabo alimentador para dentro das residências.

O sistema de TV a cabo padrão utilizava cabos coaxiais em toda a extensão da rede. Devido às altas taxas de atenuação de sinais e o grande número de amplificadores, a comunicação numa CATV padrão era unidirecional. Os sinais de vídeo eram transmitidos em *downstream* da cabeça de rede até os assinantes.

A comunicação numa CATV padrão é unidirecional.

Rede HFC

A segunda geração de rede de TV a cabo ficou conhecida como rede **Hybrid Fiber Coaxial (HFC)**. Essas redes utilizam uma combinação híbrida de fibra óptica e cabo coaxial. O meio de transmissão do centro de distribuição de TV a cabo até os pontos de conversão de mídia, denominados **transceiver ópticos**, é a fibra óptica. Partindo do *transceiver* até a conexão final do usuário, o meio utilizado passa a ser o cabo coaxial. A Figura 9.6 ilustra esquematicamente uma rede HFC.

A **cabeça de rede regional (regional cable head)** normalmente serve a 400.000 assinantes. As RCHs alimentam os **hubs de distribuição** e cada hub leva o sinal até 40.000 assinantes. Nessa nova infra-estrutura, o *hub* de distribuição assume um papel fundamental. Os esquemas de modulação e distribuição de sinais são realizados nele. Os sinais alimentam então os conver-

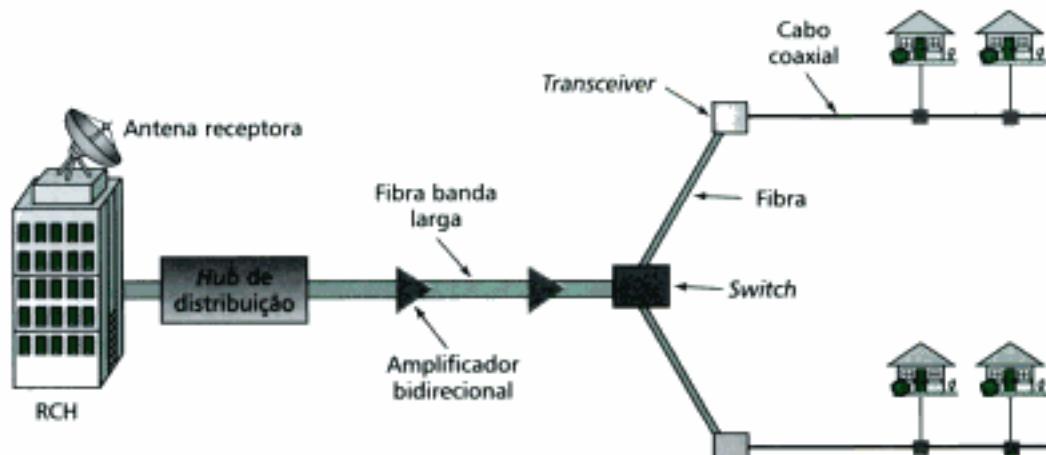


Figura 9.6 Rede HFC.

sores de mídia através dos cabos de fibra óptica. Os *transceivers* retransmitem os sinais analógicos recebidos da fibra óptica para cada segmento de cabo coaxial. Um segmento desses pode comportar até 1000 assinantes. O uso da fibra óptica reduziu drasticamente o uso de amplificadores para oito ou menos.

Uma razão da migração observada da infra-estrutura padrão da CATV para a rede HFC é a capacidade dessa rede de prover comunicação bidirecional.

A comunicação numa rede CATV/HFC pode ser bidirecional.

Largura de banda

Até mesmo num sistema HFC, a última parte da rede ainda é feita por meio metálico, no caso o cabo coaxial. Este segmento de cabo coaxial tem uma largura de banda cobrindo a faixa de 5 a 750 MHz (aproximadamente). A operadora de TV a cabo divide esta banda em três: vídeo, dados em *downstream* e dados em *upstream* (veja a Figura 9.7).

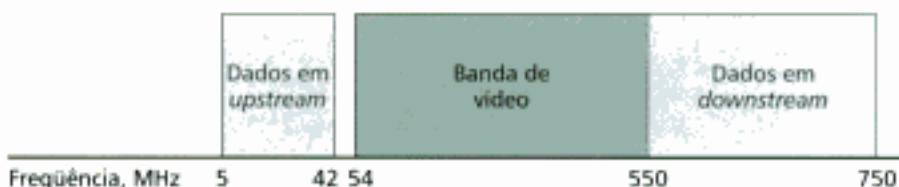


Figura 9.7 Bandas do cabo coaxial.

Banda de Vídeo

Seguindo na direção *downstream*, a **banda de vídeo** ocupa a faixa de freqüências de 54 a 550 MHz. Visto que cada canal de TV ocupa 6 MHz da banda, é possível acomodar mais de 80 canais com esse esquema.

Banda de Dados em Downstream

Os dados em *downstream*, para conexões com a Internet, ocupam a banda superior que vai de 550 a 750 MHz. Esta banda é dividida em canais de 6 MHz.

Modulação Os dados em sentido *downstream* são modulados através da técnica de modulação 64-QAM ou, possivelmente, a 256-QAM.

Os dados no sentido downstream são modulados através da técnica de modulação 64-QAM.

Taxa de Dados Na técnica 64-QAM são utilizados 6 bits para cada símbolo enviado. Nesse esquema, um bit é utilizado para correção de erros, restando 5 bits de dados/modulação. O padrão especifica 1 Hz/baud de taxa de modulação. Isto significa que, teoricamente, os dados em *downstream* podem ser recebidos a 30 Mbps (5 bits/Hz × 6 MHz). Entretanto, a taxa em *downstream* padronizada foi de 27 Mbps. Visto que o *cable modem* é conectado ao computador utilizando o padrão Ethernet 10BaseT (veja Capítulo 14), o limite de transmissão real cai para 10 Mbps.

A taxa de transmissão teórica em downstream é 30 Mbps.

Banda de Dados em Upstream

Os dados em *upstream*, para conexões com a Internet, ocupam a banda inferior que se entende de 5 a 42 MHz. Esta banda é dividida em canais de 6 MHz.

Modulação A banda de dados em *upstream* utiliza freqüências bastante susceptíveis a ruídos e interferências. Por isso, a técnica QAM não é indicada nessa banda. A melhor solução encontrada foi a QPSK.

Os dados em *upstream* são modulados utilizando a técnica de modulação QPSK.

Taxa de Dados Na modulação QPSK são utilizados 2 bits por símbolo. Novamente, o padrão especifica 1Hz/baud. Isto significa que, teoricamente, os dados em *upstream* podem ser enviados a 12 Mbps ($2\text{ bits/Hz} \times 6\text{ MHz}$). Entretanto, a taxa de transmissão real é menor que 12 Mbps.

O limite teórico da taxa de dados em *upstream* é 12 Mbps.

Compartilhamento

Tanto a banda em *upstream* quanto a banda em *downstream* pode ser compartilhada entre os assinantes.

Compartilhamento em *Upstream*

A banda disponível de dados em *upstream* é somente 37 MHz. Isto significa que existem somente seis canais de 6 MHz na direção de subida (*upstream*) de dados. O fato é: um assinante precisa usar um canal para enviar dados na direção *upstream*. A questão é: como podemos compartilhar seis canais numa região com 1000, 2000 ou, até mesmo, 100.000 assinantes? A resposta é compartilhá-la no tempo. Assim, a banda é dividida em seis canais via FDM. Estes canais devem ser compartilhados entre os assinantes numa mesma vizinhança. O provedor da CATV aloca um canal, estatística ou dinamicamente, a um grupo de assinantes. Se um assinante desejar enviar dados, ela ou ele disputa o canal com os outros que querem obter acesso; desse modo, o assinante deve esperar até o canal ficar disponível para iniciar a transmissão de dados.

Compartilhamento em *Downstream*

Temos uma situação bastante parecida na direção *downstream*. A banda disponível de dados em *downstream* possui 33 canais de 6 MHz. Um provedor de CATV tem possivelmente muito mais de 33 assinantes. Logo, outra vez os canais devem ser compartilhados entre um grupo de assinantes. Porém, a situação passa a ser diferente da anterior, pois temos a possibilidade de ocorrência de transmissões em multidifusão (*multicasting*). Se houver dados para algum dos assinantes no grupo, os dados serão enviados para aquele canal. Desde que cada assinante guarda consigo um endereço registrado no provedor, o *cable modem* para o grupo encontra o endereço do destinatário no pacotes de dados, que foi atribuído pelo provedor, e realiza a entrega, se o endereço for encontrado. De outro modo, os dados são descartados.

CM e CMTS

Para utilizar uma rede CATV na transmissão de dados precisamos de duas peças chaves: um CM e um CMTS.

CM

O *cable modem* (CM) é instalado na residência de um assinante. De certo modo, ele se parece com um modem ADSL. A Figura 9.8 ilustra o local onde este dispositivo é instalado.

CMTS

O **Cable Modem Transmission System (CMTS)** é instalado dentro do *hub* de distribuição pela provedora de TV a cabo. Ele recebe dados da Internet e os passa a um circuito combinador que os envia ao assinante. O CMTS também recebe dados do assinante e os transfere para a Internet. A Figura 9.9 mostra a localização de um CMTS.

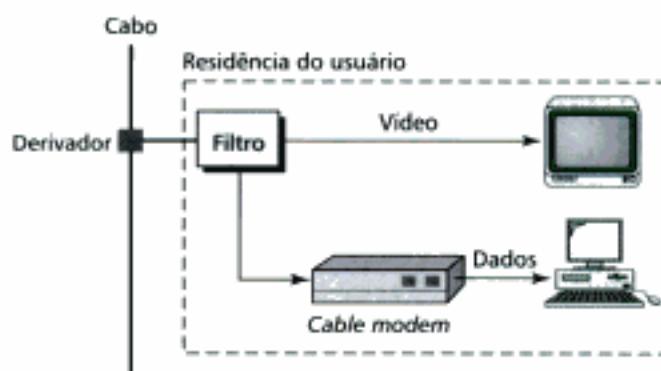


Figura 9.8 Cable modem.



Figura 9.9 CMTS.

Esquemas de Transmissão de Dados: DOCSIS

Nas últimas décadas surgiram muitos esquemas desenvolvidos especialmente para criar um padrão para as transmissões de dados em redes HFC. O esquema que prevaleceu foi o desenvolvido pela Multimedia Cable Network Systems (MCNS), conhecido como **Data Over Cable System Interface Specification (DOCSIS)**. O DOCSIS define todos os protocolos necessários ao transporte de dados entre um CMTS e um CM.

Comunicação em *Upstream*

O esquema a seguir é uma versão muito simplificada do protocolo definido pelo DOCSIS para comunicação em *upstream*. Ele descreve os passos que devem ser seguidos por um CM:

1. O CM verifica os canais em *downstream* à procura de um pacote de dados periódico, especificamente enviado pelo CMTS. Esse pacote pergunta a um novo CM se ele deseja utilizar um canal específico em *upstream*.
2. O CMTS envia um pacote ao CM definindo a localização dos canais em *downstream* e *upstream*.
3. O CM inicia então um processo, denominado **ranging**, para determinar a distância entre o CM e CMTS. Este processo é necessário para que ocorra sincronização entre eles dentro dos **minislots** usados no compartilhamento temporal do canal de *upstream*. Aprenderemos mais sobre o compartilhamento no tempo quando estivermos estudando os protocolos de contenção no Capítulo 13.
4. O CM envia um pacote ao ISP, pedindo um endereço de Internet válido.
5. O CM e o CMTS trocam alguns pacotes de maneira a estabelecer alguns parâmetros de segurança, necessários quando estamos utilizando uma rede pública como a CATV.

6. O CM envia um único identificador ao CMTS.
7. A comunicação em *upstream* pode ser iniciada no canal alocado; depois dessa sequência protocolar, o CM pode disputar os *minislots* para enviar dados.

Comunicação em *Downstream*

A comunicação é muito mais simples na direção de *downstream*. Não existe contenção porque somente uma das pontas envia dados: o CMTS. Nesse caso, o CMTS envia pacotes de dados com o endereço do CM receptor, usando o canal de *downstream* alocado.

9.3 SONET

A largura de banda espetacular dos cabos de fibra óptica é bastante apropriada às modernas tecnologias de transmissão de dados (como a vídeo conferência) que requerem altas taxas de dados e para transportar, ao mesmo tempo, um número grande de tecnologias que operam em velocidades menores. Por este motivo, a importância das fibras ópticas cresce em conjunção com o desenvolvimento de tecnologias que requerem altas taxas de transmissão de dados ou banda larga para transmissão. Como qualquer outra tecnologia foi necessário criar uma padronização. A ANSI padronizou a chamada **Synchronous Optical Network (SONET)**. O ITU-T padronizou a hierarquia síncrona digital (*Synchronous Digital Hierarchy – SDH*)*. Os dois padrões são bastante parecidos.

Dentre as diversas características atribuídas a SONET e a SDH, três são particularmente interessantes para nós.

Primeiro, a SONET é uma rede síncrona. Um relógio (*clock*) único é utilizado para controlar a temporização das transmissões e os equipamentos dentro da rede como um todo. Uma sincronização desse tipo adiciona um nível de previsibilidade ao sistema. Esta característica aliada aos *frames* especialmente desenvolvidos para as aplicações da rede, habilita cada canal individual a ser multiplexado, o que resulta em melhorias em termos de velocidade e redução dos custos da rede.

Segundo, a SONET traz nativamente os padrões recomendados para os sistemas de transmissão por fibra óptica (*Fiber-Optic Transmission System – FOTS*), equipamentos vendidos por diversos fabricantes diferentes.

Terceiro, as especificações físicas da SONET e o uso dos *frames* específicos incluem mecanismos capazes de transportar sinais de sistemas tributários incompatíveis (tal como DS-0 e DS-1). É nesta flexibilidade que reside a reputação de conectividade universal da SONET.

A SONET é um excelente exemplo de sistema de multiplexação por divisão de tempo (TDM). A banda da fibra é considerada como um único canal dividido em *time-slots* de modo a definir subcanais. A SONET, como uma rede TDM, é um sistema síncrono controlado por um relógio (*clock*) mestre com um alto nível de precisão. A transmissão de *bits* é controlada pelo relógio mestre.

A SONET é um sistema TDM síncrono controlado por um relógio mestre.

Dispositivos SONET

A transmissão SONET funciona com base em três dispositivos: multiplexadores/demultiplexadores **Synchronous Transport Signal (STS)**, regeneradores e multiplexadores *add/drop*. A Figura 9.10 mostra um exemplo de uma SONET.

- **Multiplexador/Demultiplexador STS.** Um Mux/Demux STS multiplexa os sinais de múltiplas fontes ou demultiplexa um sinal STS em diferentes fontes de sinais.

* N. de R. T.: Adotada no Brasil como padrão.

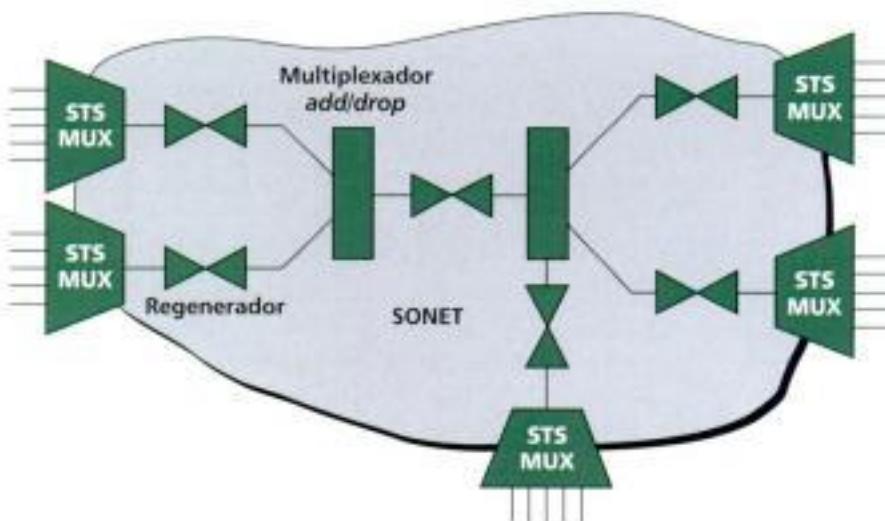


Figura 9.10 Uma SONET.

- **Regenerador.** Um **regenerador STS** é um repetidor (veja Capítulo 16) que recebe um sinal óptico e o regenera até o nível original. Entretanto, os regeneradores neste sistema adiciona uma função que os repetidores da camada física não dispõem. Um regenerador SONET substitui parte da informação de *overhead* (informação de cabeçalho) por uma nova informação. Assim, estes dispositivos operam na camada de enlace, contrariamente aos repetidores ordinários.
- **Multiplexador add/drop.** Um **multiplexador add/drop** pode adicionar sinais provenientes de diferentes fontes a um caminho (rota) predeterminada ou remover um sinal desejado de um caminho (rota) e redirecioná-lo, sem demultiplexar todo o sinal.

Frame SONET

Um *frame* SONET deve ser visualizado como uma matriz de nove linhas, o qual contém 90 octetos (*bytes*), perfazendo um total de 810 octetos na matriz (veja a Figura 9.11). Alguns desses octetos são usados para funções de controle; eles não estão posicionados no início ou no fim do *frame* (como a maioria dos cabeçalhos de *frames*).

As três primeiras colunas do *frame* são usadas para administração do *overhead* (sinalização). O restante do *frame* é conhecido como **Synchronous Payload Envelope (SPE)**. O SPE contém a



Figura 9.11 Formato do *frame*.

transmissão de *overhead* e os dados do usuário. O *payload* porém não inicia necessariamente na linha 1 coluna 4. Ele pode começar em qualquer lugar dentro do *frame* e pode até se estender a dois *frames*. Esta característica permite bastante flexibilidade; se o SPE atrasar por algum motivo, após o *frame* ter iniciado, ele não tem que esperar até o início do próximo *frame*. Um ponteiro (endereço) ocupando as colunas de 1 a 3 (linha 4) é capaz de determinar o endereço de início (linha e coluna) do SPE.

Transmissão de Frames

A SONET define uma hierarquia de níveis de serviço denominada *synchronous transport signals* (STS). Cada STS (STS-1 a STS-192) suporta uma taxa de dados específica, dada em *megabits* por segundo (veja a Tabela 9.1). O link físico definido para transportar cada nível de STS é conhecido como ***Optical Carriers*** (OC). Os níveis OC descrevem as especificações físicas e conceituais do suporte de cada nível de serviço STS. A implementação real destas especificações é deixada aos fabricantes. Atualmente, as implementações mais comuns são a OC-1, OC-3, OC-12 e OC-48.

TABELA 9.1 Taxas da SONET

STS	OC	Taxa original (Mbps)	SPE (Mbps)	Usuário (Mbps)
STS-1	OC-1	51,84	50,12	49,536
STS-3	OC-3	155,52	150,336	148,608
STS-9	OC-9	466,56	451,008	445,824
STS-12	OC-12	622,08	601,344	594,432
STS-18	OC-18	933,12	902,016	891,648
STS-24	OC-24	1244,16	1202,688	1188,864
STS-36	OC-36	1866,23	1804,032	1783,296
STS-48	OC-48	2488,32	2405,376	2377,728
STS-192	OC-192	9953,28	9621,604	9510,912

STS-1

A STS-1 ou OC-1 é o serviço de menor taxa provido pela SONET. A STS-1 transmite 8000 *frames*/s. A Figura 9.12 compara a taxa SPE original com a taxa do usuário. As taxas refletem o número de colunas disponíveis no *frame*. Por exemplo, a taxa de transmissão SPE é menor que a taxa de transmissão original devido às três colunas de gerenciamento do *frame*.

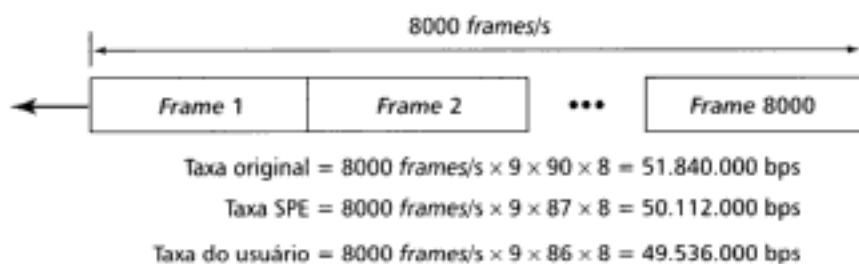


Figura 9.12 Taxa de dados.

Tributários Virtuais

A SONET foi desenvolvida para transportar massas de dados em banda larga. Entretanto, as taxas de dados da hierarquia digital atual (DS-1 a DS-3) são menores que a STS-1. Para tornar a SONET compatível com a hierarquia atual, os *frames* foram montados de modo a incluir um sistema de **tributários virtuais (VTs)**. Um tributário virtual é um *payload* parcial (entretanto útil) para preen-

chimento do *frame*. Em vez de usar todas as 87 colunas de um frame SPE para dados de uma única fonte, podemos subdividi-la e denotar a cada componente um VT.

Quatro tipos de VTs foram definidos de modo a se adaptar às hierarquias existentes (veja Figura 9.13). Note que o número de colunas permitidas para cada tipo de VT pode ser determinado dobrando-se o número de identificação da VT (VT1.5 recebe três colunas, a VT2 recebe quatro colunas, etc.).

- **VT1.5.** O VT1.5 se adapta ao serviço americano DS-1 (1.544 Mbps).
- **VT2.** O VT2 se adapta ao serviço europeu CEPT-1 (2.048 Mbps).
- **VT3.** O VT3 se adapta ao serviço DS-1C (DS-1 fracionário; 3.152 Mbps)
- **VT6.** O VT6 se adapta ao serviço DS-2 (6.312 Mbps).

Quando dois ou mais tributários são inseridos num único *frame* STS-1, eles são intercalados coluna por coluna. A SONET possui um mecanismo de identificação de cada VT e os separa sem demultiplexar todo o sistema. A discussão destes mecanismos e as formas de controle atrás deles está fora do escopo deste livro.

$$\begin{aligned} \text{VT1.5} &= 8000 \text{ frames/s} \times 3 \text{ colunas} \times 9 \text{ linhas} \times 8 \text{ bits} = 1,728 \text{ Mbps} \\ \text{VT2} &= 8000 \text{ frames/s} \times 4 \text{ colunas} \times 9 \text{ linhas} \times 8 \text{ bits} = 2,304 \text{ Mbps} \\ \text{VT3} &= 8000 \text{ frames/s} \times 6 \text{ colunas} \times 9 \text{ linhas} \times 8 \text{ bits} = 3,456 \text{ Mbps} \\ \text{VT6} &= 8000 \text{ frames/s} \times 12 \text{ colunas} \times 9 \text{ linhas} \times 8 \text{ bits} = 6,912 \text{ Mbps} \end{aligned}$$

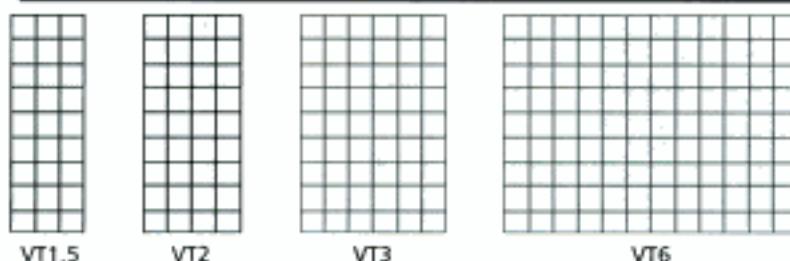


Figura 9.13 Tipos de VTs.

Serviços em Taxas Elevadas

Os sistemas de STS de taxas menores podem ser multiplexados de modo a torná-los compatíveis com o sistema de taxas mais altas. Por exemplo, três STS-1 podem ser combinados para formar um STS-3; quatro STS-3 multiplexados formam um STS-12 e assim por diante. A Figura 9.14 mostra como podemos multiplexar três STS-1 e formar um STS-3. Para criar um STS-12, a partir de serviços mais lentos, podemos multiplexar 12 STS-1 ou 4 STS-3.

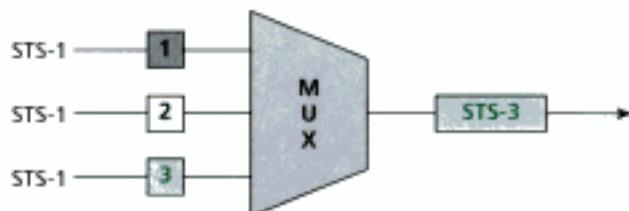


Figura 9.14 Multiplexação STS.

9.4 TERMOS-CHAVE

<i>Asymmetric DSL (ADSL)</i>	Multiplexador <i>add/drop</i>
Banda de dados em <i>downstream</i>	Multiplexador/Demultiplexador STS
Banda de dados em <i>upstream</i>	Multiplexador de acesso DSL (DSLAM)
Banda de vídeo	<i>Optical Carrier (OC)</i>
Cabeça de rede (<i>head end</i>)	<i>Ranging</i>
Cabeça de Rede Regional (RCH)	Regenerador
<i>Cable Modem (CM)</i>	Symmetric DSL (SDSL)
Cable Modem Transmission System (CMTS)	Synchronous Optical Network (SONET)
<i>Discrete Multitone Technique (DMT)</i>	Synchronous Payload Envelope (SPE)
Hierarquia Síncrona Digital (<i>Synchronous Digital Hierarchy – SDH</i>)	Synchronous Transport Signal (STS)
High-bit-rate DSL (HDSL)	<i>Transceiver</i>
<i>Hub</i> de distribuição	Tributário Virtual (VT)
Hybrid Fiber-Coaxial network (HFC)	TV a cabo (CATV)
<i>Minislot</i>	Very-high-bit-rate DSL (VDSL)

9.5 RESUMO

- Um computador pessoal pode acessar a Internet através da rede de telefonia pública ou através do sistema de TV a cabo.
- A tecnologia DSL suporta velocidades altas de comunicação de dados e utiliza a rede de telefonia pública.
- A tecnologia ADSL permite que consumidores residenciais acessem a Internet a taxas de transmissão de até 1 Mbps na direção de *upstream* (subida) e até 8 Mbps na direção de *downstream* (descida).
- ADSL utiliza uma técnica de modulação conhecida como DMT, a qual combina QAM e FDM.
- SDSL, HDSL e VDSL são outras tecnologias DSL.
- Teoricamente, o cabo coaxial utilizado nas transmissões de TV a cabo permitem acessar a Internet em taxas de transmissão até 12 Mbps na direção *upstream* e 30 Mbps na direção *downstream*.
- Uma rede HFC permite que usuários do sistema CATV acessem a Internet através de uma combinação de cabos formada por fibras ópticas e cabos coaxiais.
- A banda do cabo coaxial é dividida numa banda de vídeo, uma banda de dados em *downstream* e um banda de dados em *upstream*. Tanto a banda em *upstream* quanto a banda em *downstream* são compartilhadas entre os assinantes.
- DOCSIS define todos os protocolos necessários à transmissão de dados dentro de uma rede HFC.
- Synchronous Optical Network (SONET) é uma rede TDM síncrona de elevada capacidade de transmissão de dados utilizando redes de fibras ópticas.
- SONET possui uma hierarquia de sinais definida (parecida com a hierarquia DS) conhecida como Synchronous Transport Signals (STS).
- Os níveis de Optical Carrier (OC) são as implementações físicas das STS's.
- Um *frame* SONET pode ser visualizado como uma matriz de 9 linhas, contendo 90 octetos cada.
- Um sistema SONET pode usar os seguintes equipamentos:
 - Multiplexador STS – combina muitos sinais ópticos de modo a construir um sinal STS.
 - Regenerador – remove o ruído do sinal óptico.
 - Multiplexador *add/drop* – adiciona os STS's de diferentes caminhos e remove STS's de um caminho.
- A SONET é compatível com a hierarquia atual, os *frames* foram montados de modo a incluir o conceito de sistema de tributários virtuais (VTs). Os VTs são *payloads* parciais consistindo de um bloco $m \times n$ de octetos. Um *payload* STS pode ser formado de uma combinação de VTs.
- STSs podem ser multiplexados de modo a estabelecer um novo STS de capacidade maior.

9.6 PRATIQUE OS CONHECIMENTOS ADQUIRIDOS

Questões de Revisão

1. Cite duas tecnologias que possuem taxa de transmissão mais altas que os modems analógicos.
2. Por que a ADSL não está disponível para empresas? Que tecnologia DSL atende melhor as expectativas das empresas?
3. Quem são os usuários principais da tecnologia ADSL?
4. Como os filtros limitam a largura de banda de uma conexão local?
5. Qual é a técnica de modulação utilizada na tecnologia ADSL?
6. Quais são os tipos de dispositivos ADSL necessários a um usuário?
7. Qual é o propósito de um DSLAM?
8. De que modo o HDSL é superior a uma linha T-1?
9. Qual é a função da cabeça de rede numa rede de TV a cabo?
10. Descreva o meio de transmissão numa rede HFC.
11. Por que a modulação QAM não é utilizada no canal de dados *upstream* de uma rede HFC?
12. Qual a diferença entre CM e CMTS?
13. Qual é o propósito do DOCSIS?
14. Qual é a diferença entre os multiplexadores STS e *add/drop*?
15. Qual é o tipo de relação entre os níveis STS e OC?
16. Qual é a relação entre a rede SONET e a hierarquia digital síncrona (SDH)?
17. Por que a rede SONET é denominada síncrona?
18. Qual é a função de um regenerador SONET?
19. Como um *frame* STS-1 é organizado?
20. O que é um tributário virtual?
21. De que maneira podemos tornar os sistemas STS mais lentos compatíveis com os sistemas STS mais rápidos?

Questões de Múltipla Escolha

22. _____ possui uma capacidade de transmissão na direção de *downstream* maior que a taxa na direção *upstream*.
 - VDSL
 - ADSL
 - SDSL
 - (a) e (b)
23. _____ está disponível para serviços das empresas que requerem taxas de dados uniformes nas duas direções.
 - VDSL
 - ADSL
 - SDSL
 - (a) e (b)
24. Os _____ limitam a largura de banda da conexão telefônica local a 4 kHz.
 - Transceivers*
 - Filtros
 - Repetidores
 - Hubs
25. DMT é uma técnica de modulação que combina elementos da _____ e _____.
 - FDM; TDM
26. A porção mais alta da banda ADSL transporta _____.
 - Comunicação de voz
 - Dados em *upstream*
 - Dados em *downstream*
 - Controle de dados
27. A taxa de dados real de uma conexão em *downstream* ADSL é _____.
 - 64 kbps a 1 Mbps
 - 6 a 30 kbps
 - 31 kbps a 255 Mbps
 - 500 kbps a 8 Mbps
28. _____ é um dispositivo que a empresa telefônica utiliza para empacotar os dados a serem enviados ao servidor ISP.
 - Um DSLAM
 - Um modem ADSL
 - Um filtro
 - Um divisor
29. _____ foi desenvolvido como uma alternativa à linha T-1.
 - VDSL

- b. ADSL
c. SDSL
d. HDSL
30. A tecnologia HDSL codifica dados usando _____.
 a. 4B/5B
b. 2B1Q
c. 1B2Q
d. 6B/8T
31. Um sinal codificado em _____ está mais suscetível à atenuação que um sinal codificado em _____.
 a. AMI; 2B2Q
b. 2B1Q; AMI
c. AMI; 2B1Q
d. Nenhuma das anteriores
32. Outro nome para a central de distribuição da CATV é _____.
 a. Divisor
b. Transceiver
c. Combinador
d. Cabeça de rede
33. A TV a cabo padrão transmite sinais _____.
 a. Upstream
b. Downstream
c. Nas duas direções
d. Nenhuma das anteriores
34. Uma rede HFC usa _____ como meio de transmissão do switch até os transceivers.
 a. Fibra ótica
b. Cabo coaxial
c. UTP
d. STP
35. Numa rede HFC, o hub de distribuição realiza a _____ dos sinais.
 a. Modulação
b. Distribuição
c. Divisão
d. (a) e (b)
36. Um canal de TV numa rede HFC precisa de uma banda de _____ MHz.
 a. 6
b. 100
c. 250
d. 369
37. Dados em _____ vão da rede do assinante até a Internet.
 a. Upstream
- b. Downstream
c. Midstream
d. Nenhuma das anteriores
38. Numa rede HFC, os dados em upstream são modulados usando a técnica de modulação _____.
 a. QAM
b. QPSK
c. PCM
d. ASK
39. O padrão para a transmissão de dados nas redes HFC é chamado _____.
 a. MCNS
b. DOCSIS
c. CMTS
d. ADSL
40. O _____ é um dispositivo de rede HFC instalado normalmente dentro do hub de distribuição que recebe dados da Internet e os repassa ao circuito combinador.
 a. CM
b. CMTS
c. DCOSIS
d. MCNS
41. SONET é um padrão para redes de _____.
 a. Par trançado
b. Cabo coaxial
c. Ethernet
d. Fibra óptica
42. SONET é um acrônimo para _____ NETwork.
 a. Synchronous Optical
b. Standard Optical
c. Symmetric Open
d. Standard Open
43. Num sistema SONET, um _____ pode remover sinais de um caminho na rede.
 a. Multiplexador STS
b. Regenerador
c. Multiplexador add/drop
d. Repetidor
44. O sistema SPE de um frame STS-1 contém _____.
 a. Ponteiros
b. Dados do usuário
c. Overhead
d. (b) e (c)

Exercícios

45. Mostre como multiplexadores STS-9 podem ser organizados de modo a formar um STS-36. Há algum tipo de *overhead* extra neste tipo de multiplexação? Por quê?
46. Qual é o tempo de duração de um *frame* STS-1?
47. Qual é o tempo de duração de um *frame* STS-3, STS-9,..., STS-192?
48. Quantos VT1.5's podem ser transportados num único *frame* STS-1?
49. Quantos VT2's podem ser transportados num único *frame* STS-1?
50. Quantos VT3's podem ser transportados num único *frame* STS-1?
51. Quantos VT6's podem ser transportados num único *frame* STS-1?
52. Um usuário deseja enviar dados a 3 Mbps. Que VT ou combinação de VTs poderiam ser utilizadas?
53. Um usuário deseja enviar dados a 7 Mbps. Que VT ou combinação de VTs poderiam ser utilizadas?
54. Um usuário deseja enviar dados a 12 Mbps. Que VT ou combinação de VTs poderiam ser utilizadas?
55. Que VT permite uma taxa de dados idêntica à linha T-1?
56. Que VT ou STS transmite(m) quase a mesma taxa de dados que uma linha T-3?
57. Um empresa deseja utilizar a SONET para multiplexar cerca de 100 canais de voz digitalizados. Que VT ou combinação de VTs podem ser adotados por esta empresa?
58. Desenhe uma SONET usando todos os seguintes dispositivos. Rotele todas as linhas, sessões e caminhos.
 - a. Três multiplexadores STS (dois como entrada e um como saída)
 - b. Quatro multiplexadores *add/drop*
 - c. Cinco regeneradores

PARTES

CAMADA DE ENLACE DE DADOS

A camada de enlace de dados está localizada entre a camada de rede e a camada física no modelo da Internet. Ela recebe os serviços da camada física e provê serviços para a camada de rede. A Figura 1 mostra a posição da camada de enlace de dados na arquitetura da Internet.

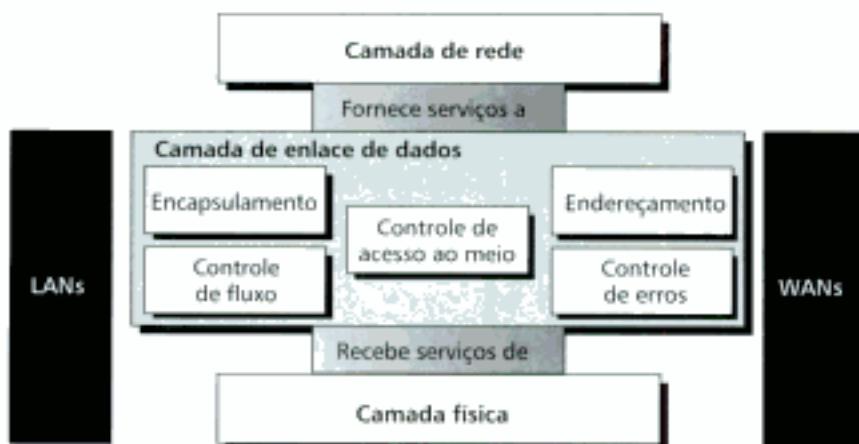


Figura 1 Posição da camada de enlace de dados.

A **camada de enlace de dados** é responsável por transportar pacotes de um nó (computador ou roteador) a outro através da rede. Diferentemente da camada de rede que desempenha um papel global na arquitetura, a camada de enlace tem um nível de responsabilidade apenas local. Todos os processos envolvendo dois nós é de responsabilidade da camada de enlace. Noutras palavras, visto que LANs e WANs são delimitadas por nós de rede, podemos dizer que a responsabilidade da camada de enlace é encaminhar pacotes através de uma LAN ou WAN.

A integridade dos pacotes deve ser preservada durante a viagem através de uma LAN ou WAN (entre dois nós). A camada de enlace deve prover mecanismos que assegurem integridade aos pacotes das camadas superiores do modelo. Se um pacote for corrompido durante uma trans-

missão, a camada de enlace deve ser capaz de corrigi-lo ou pedir retransmissão desse pacote. A camada de enlace deve também assegurar que o próximo nó da rede não está sendo inundado com os dados provenientes do nó anterior, isto é, essa camada tem que prover controle do fluxo de dados.

Acessar uma LAN ou uma WAN para enviar dados é também outra questão a ser tratada no nível da camada de enlace. Se muitos computadores ou roteadores estiverem conectados num mesmo *link* (meio) e mais de um desses dispositivos solicitar o envio de dados ao mesmo tempo, qual deles irá receber o direito à transmissão? Qual é o método de acesso ao meio?

Serviços da Camada de Enlace de Dados

Os serviços da camada de enlace incluem encapsulamento/desencapsulamento de dados, endereçamento, controle de erro, controle de fluxo e controle de acesso ao meio (veja Figura 2).



Figura 2 Serviços da camada de enlace de dados.

Encapsulamento/Desencapsulamento

A camada de enlace de dados é responsável pelo fluxo de dados entre dois nós adjacentes. Para chegar ao próximo nó, os dados devem atravessar uma LAN ou uma WAN, cada qual operando de acordo com os protocolos da rede em questão. Os pacotes oriundos da camada superior devem assim ser empacotados de modo apropriado na camada de enlace da LAN ou WAN onde estiver ocorrendo o processo. Diferentes protocolos tratam os pacotes da camada de enlace com diferentes nomes. Entretanto, na maioria das LANs os pacotes são denominados *frames* ou quadros. Nas WANs ATM, um pacote da camada de enlace recebe o nome de célula (*cell*). Veremos exemplos de encapsulamentos nos Capítulos 12-18.

Endereçamento

É necessário que a camada de enlace forneça um mecanismo de endereçamento. O endereçamento na camada de enlace recebe o nome de endereço físico ou endereço MAC e é utilizado para determinar o endereço do próximo nó no processo de entrega entre dois nós (*hop-to-hop*). O endereço físico utilizado dentro de uma LAN é totalmente diferente do endereço físico usado na WAN. Numa LAN, o dispositivo que tiver dados a transmitir utiliza o endereço do próximo nó para enviar um *frame* através dessa LAN. Uma WAN geralmente utiliza o endereço de um circuito virtual para essa finalidade. Discutiremos os mecanismos de endereçamento nos Capítulos 14 a 18.

Controle de Erros

Os erros são inevitáveis na comunicação de dados. Escolhendo-se um bom equipamento disponível no mercado e um meio de transmissão mais confiável podemos reduzir a freqüência de ocorrência dos erros, mas nunca poderemos eliminá-los. As redes devem ser dotadas da capacidade de transmitir dados entre dispositivos nela conectados com total precisão. Introduziremos as técnicas de controle de erros no Capítulo 10. Em seguida, no Capítulo 11, discutiremos o controle de erros como parte integrante (nativa) da camada de enlace de dados.

Controle de Fluxo

Outro item de responsabilidade da camada de enlace é o controle de fluxo. Na maioria dos protocolos, o controle de fluxo é um conjunto de procedimentos que ensinam o dispositivo que tem dados a transmitir, a quantidade (a massa) de dados que ele pode transmitir sem ter que esperar um *ack* (*acknowledgment* – confirmação) do receptor. O controle de fluxo não deve permitir que o receptor seja inundado com os dados provenientes do transmissor. O dispositivo receptor deve ser capaz de informar ao dispositivo transmissor que diminua, ou até mesmo, que pare de enviar *frames* antes que algum limite de capacidade seja atingido no *buffer* de recepção. Discutiremos os mecanismos de controle de fluxo como parte integrante (nativa) da camada de enlace no Capítulo 11.

Controle de Acesso ao Meio (*Medium Access Control* – MAC)

Quando computadores compartilham um meio (cabo ou ar), algum tipo de mecanismo operacional deve controlar o dispositivo que acessa esse meio num dado momento. Para prevenir conflitos ou colisões numa rede é necessário um método de controle de acesso ao meio (MAC). Este método define o procedimento a ser seguido por um computador quando ele necessitar enviar um ou mais *frames*. Dedicaremos dois capítulos à essa questão, Capítulos 12 e 13.

Redes Locais (*Local Area Networks* – LANs)

As Redes Locais (LANs) operam nas camadas física e de enlace de dados. Assim, o lugar óbvio onde devemos discutir as redes LANs é após a discussão dessas duas camadas. Dedicamos o Capítulo 14 aos padrões coletivamente denominados Ethernet, o padrão de redes LAN mais comum hoje em dia, e o Capítulo 15 às LANs sem fio (*Wireless LANs*), a maior aposta no futuro das LANs. Tratando esses dois assuntos, mostraremos como conectar LANs no Capítulo 16.

Padrões IEEE

A Internet não impõe especificações para LANs e WANs. Ao contrário, ela aceita qualquer padrão de rede local (LAN) como rota de comunicação dos pacotes provenientes da camada de rede. O fato básico é que coexistem muitos protocolos de controle das LANs. Em 1985, a Computer Society do IEEE iniciou um projeto, denominado Projeto 802 (*802 Project*), para estabelecer padrões e permitir a interconectividade entre equipamentos de diversos fabricantes. A IEEE subdividiu a camada de enlace em duas subcamadas: controle do link lógico (*Logical Link Control* – LLC) e controle de acesso ao meio (*Medium Control Access* – MAC), conforme Figura 3. A subcamada LLC não é uma arquitetura específica propriamente dita, isto é, ela é a mesma para toda as definições de LANs do IEEE. Assim, hoje em dia, ela não é tão amplamente utilizada. Por outro lado, a subcamada MAC possui uma quantidade enorme de padronizações diferentes, cada qual contendo informações específicas, proprietárias do produto LAN que estiver sendo utilizado. A Figura 4 mostra alguns dos padrões 802 do IEEE para LANs específicas.

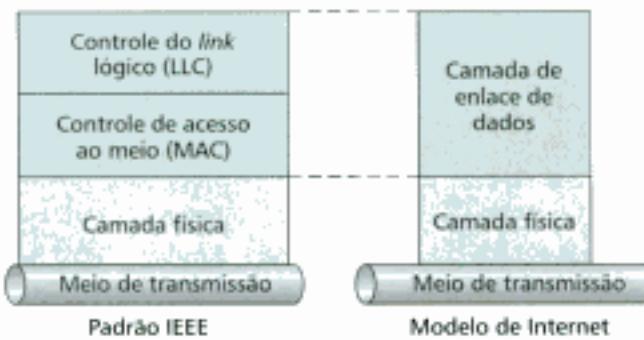


Figura 3 Subcamadas LLC e MAC.



Figura 4 Padrões IEEE para LANs.

Redes de Longa Distância (*Wide Area Networks – WANs*)

As redes WANs também operam nas camadas física e de enlace de dados e são discutidas nesta parte do texto. No Capítulo 17, estudaremos os sistemas de telefonia móvel e via satélite como WANs sem fio (*wireless WANs*). No Capítulo 18, discutiremos as redes Frame Relay e ATM como redes de comutação de WANs.

Organização dos Capítulos

A Parte III deste livro cobre nove capítulos ao todo (Capítulos 10 a 18). Nos Capítulos 10-13, estudaremos genericamente os serviços suportados pela camada de enlace de dados: controle de erro, controle de fluxo e acesso ao meio. O Capítulo 10 trata da questão da detecção de erros, um prelúdio ao controle de erros. O Capítulo 11 trata sobre fluxo e controle de erros. O Capítulo 12 explica o controle de acesso ao meio para conexões ponto a ponto. O Capítulo 13 faz o mesmo, só que para o acesso multiponto.

Os Capítulos 14 a 16 são totalmente dedicados às LANs. O Capítulo 14 trata o padrão mais difundido das redes locais: o padrão Ethernet. O Capítulo 15 discute as LANs sem fio (*wireless LANs*). Por fim, o Capítulo 16 mostra como conectar LANs.

Os Capítulos 17 e 18 são totalmente dedicados as WANs. O Capítulo 17 retrata as WANs sem fio (*wireless WANs*), redes de telefonia móvel e redes de satélites. O Capítulo 18 encerra a Parte III do livro tratando sobre comutação nas WANs, isto é, Frame Relay e ATM.

Detecção e Correção de Erros

A premissa fundamental sobre redes de dados é que elas devem ser capazes de transferir dados de um dispositivo a outro com total precisão. Um sistema que não pode ou não consegue garantir a entrega de dados, isto é, sem integridade da informação, pelo dispositivo receptor é essencialmente inútil. Além disso, toda vez que dados estiverem sendo transmitidos entre nós de uma rede eles podem ser corrompidos durante a passagem. Muitos fatores podem afetar, modificar ou destruir um ou mais *bits* de uma seqüência de dados. A maioria dos sistemas de alta confiabilidade dispõem de mecanismos para detecção e correção de erros.

Dados podem ser corrompidos durante a transmissão. Os erros devem ser detectados e corrigidos para que uma comunicação seja considerada confiável.

10.1 TIPOS DE ERROS

Sempre que se estabelece um fluxo de dados de um ponto a outro, tal fluxo está sujeito a sofrer modificações imprevisíveis provocadas pela interferência. A interferência pode modificar a forma do sinal original. Em um erro simples, ou seja, aquele onde apenas um *bit* é modificado por vez, ocorre a troca de um 0 por um 1 ou vice-versa. Por exemplo, durante uma transmissão de dados, um ruído em rajada impulsiva num intervalo de tempo de 0,01s pode modificar todos os 12 *bits* de informação de uma transmissão a 1200 bps.

Erros Isolados

A expressão **erros isolados** é aplicada sempre que apenas um *bit* da unidade de informação (tal como *byte*, caractere, seqüência de dados ou pacote) é modificado.

Em um erro isolado é modificado um único *bit* por vez na seqüência de dados.

A Figura 10.1 ilustra o efeito da incidência de um erro isolado numa certa seqüência de dados. Para compreender o impacto da mudança, imagine que cada grupo de 8 *bits* corresponde a um caractere ASCII com um *bit* 0 adicionado mais à esquerda do conjunto. Na figura, a seqüência enviada (00000010) corresponde ao caractere *STX* do código ASCII e significa *start of text* (início do

texto). Do outro lado, a figura mostra que um processo de interferência corrompeu a seqüência de dados para 00001010, que em ASCII corresponde ao caractere *LF*, isto é, *line feed* – avanço de linha (para mais informações sobre o código ASCII, veja o Apêndice A).

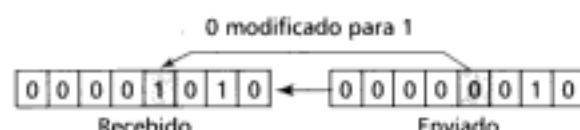


Figura 10.1 Erro isolado.

Os erros isolados são muito freqüentes numa transmissão serial de dados. Para compreender porque, imagine que a fonte envie dados a uma taxa de 1 Mbps. Isto significa que o tempo de duração de cada *bit* é $1/1.000.000 = 1 \mu s$. Para que um único *bit* seja corrompido, um ruído deve se manifestar em apenas $1 \mu s$, o que é muito raro, pois ruidos se manifestam normalmente durante um intervalo de tempo muito maior.

Contudo a transmissão serial não é a única fonte de incidência dos erros isolados. Numa comunicação paralela, um único *bit* da seqüência de dados também pode ser modificado. Por exemplo, imagine uma linha paralela para transmissão de dados a 8 fios sendo utilizada para enviar 8 *bits* ao mesmo tempo e que um dos fios está sujeito a uma interferência maior. Nesse caso, é possível que 1 *bit* seja corrompido em cada *byte* transmitido. Imagine ainda, que essa transmissão paralela ocorre dentro de um computador, entre CPU e memória, e que estejamos transmitindo 8 Mbytes de dados. Perceba como isso pode ser grave.

Rajada de Erros

O termo **rajada de erros (burst error)** deve ser utilizado sempre que 2 ou mais *bits* da seqüência de dados forem corrompidos.

Dois ou mais bits da seqüência de dados são corrompidos numa rajada de erros.

A Figura 10.2 ilustra o efeito de uma rajada de erros sobre uma certa seqüência de dados. Neste caso, o transmissor envia 0100010001000011 e o receptor recebe 0101110101000011. Note que uma rajada de erros não precisa incidir necessariamente em *bits* consecutivos da seqüência. O comprimento da rajada é medido do primeiro ao último *bit* corrompido na seqüência. Significa que, durante a medição do comprimento, alguns *bits* intermediários podem não ter sido corrompidos.

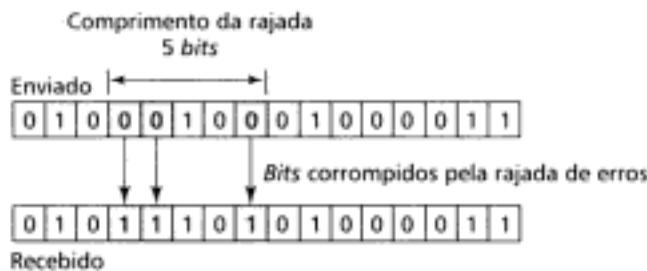


Figura 10.2 Rajada de erros com comprimento de 5 bits.

As rajadas de erros são mais freqüentes numa transmissão serial. Normalmente, o tempo de duração de um ruído é muito maior que o tempo de um *bit*. Sendo assim, quando um ruído afetar uma certa seqüência de dados, um conjunto de *bits* da seqüência podem ser corrompidos ao mesmo tempo. A quantidade de *bits* afetados na seqüência depende da taxa de transmissão de dados

e do intervalo de duração do ruído. Por exemplo, se estivermos transmitindo dados a 1 Kbps, um ruído de 1/100s pode afetar 10 bits ao mesmo tempo. Se estivermos transmitindo a 1 Mbps, o mesmo ruído pode afetar 10.000 bits.

10.2 DETECÇÃO DE ERROS

Embora o objetivo da verificação de erros leve à correção dos mesmos, na maioria das vezes, primeiramente devemos detectá-los. É muito mais simples detectar um erro do que corrigi-lo, mas é o primeiro passo no processo de correção de erros.

Redundância

Um mecanismo eficiente de **detecção de erros** seria enviar os dados duplicados. O dispositivo receptor seria então capaz de comparar bit a bit entre as duas versões de dados enviados e apontar possíveis erros. Quaisquer discrepâncias indicariam a ocorrência de erros. Assim, um mecanismo apropriado de correção poderia ser acionado para a correção efetiva dos erros. Este sistema seria totalmente preciso (a probabilidade de ocorrência de erros exatamente nos mesmos bits de ambos conjuntos de dados é infinitesimalmente pequena), mas também seria insuportavelmente lento. Não somente o tempo de transmissão seria duplicado, mas também os esforços para comparar bit a bit as duas unidades de dados.

A idéia de incluir informação extra numa transmissão para facilitar a detecção de erros foi uma excelente solução para o problema. Entretanto, em vez de repetir todo o fluxo de dados, foi adotado a inclusão de uma certa quantidade de bits adicionais no final de cada seqüência de dados. Esta técnica é denominada **redundância** porque os bits extra são informação redundante, isto é, eles são descartados tão logo a verificação da transmissão tenha sido realizada.

As detecções de erros utilizam o conceito de redundância, que é a técnica de adicionar bits extras no final da unidade de informação para facilitar a detecção de erros no destinatário.

A Figura 10.3 mostra um fluxo do processo de detecção de erros e a garantia da precisão da seqüência de dados utilizando a técnica de redundância. Uma vez que o fluxo de dados tenha sido estabelecido, ele passa através de um dispositivo que o analisa e adiciona, apropriadamente, os dados redundantes de verificação no final da unidade. Sendo assim, a seqüência de dados original, acrescida de muitos bits extra, viaja através de um link até o receptor. O receptor coloca todo o fluxo numa função de verificação de erros. Se o fluxo de bits passar pelos critérios de verificação, a porção de dados adicionados à seqüência de informação original (redundância) é descartada e os dados propriamente ditos são aceitos como íntegros.

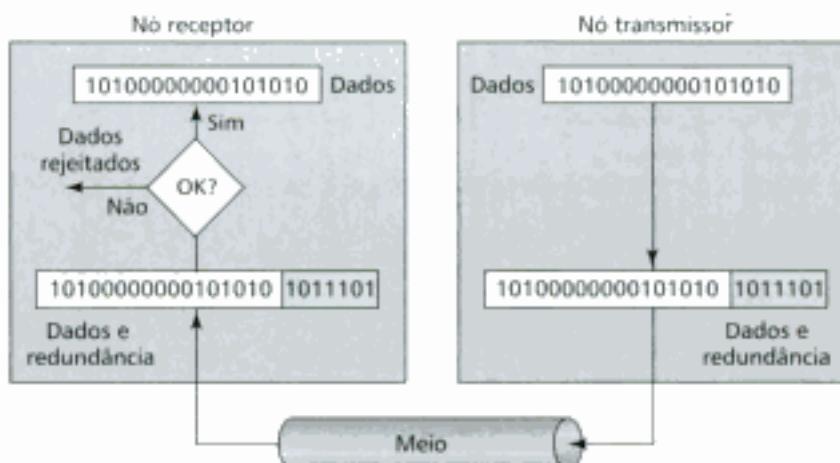


Figura 10.3 Redundância.

Três tipos de verificação de redundância são utilizados na comunicação de dados: teste de paridade, CRC (Cyclic Redundancy Check) e *checksum* (veja Figura 10.4).



Figura 10.4 Métodos de detecção.

Teste da Paridade

O **teste ou verificação da paridade**, além de ser o mecanismo de detecção de erros mais comum é também o de menor custo. Existem dois tipos de teste de paridade: paridade de caractere e paridade combinada*.

Verificação do Bit de Paridade em Caracteres

Nesta técnica, um *bit* redundante (denominado **bit de paridade**) é adicionado a cada seqüência de dados (tipicamente os caracteres) de tal modo que o número total de 1s na seqüência (incluindo o *bit* de paridade) torne-se par ou ímpar. Exemplificando, suponha que desejamos transmitir o caractere *a* em ASCII. Consultando o Apêndice A, percebemos que o caractere *a* minúsculo equivale a 97 em decimal e 1100001 em binário (veja a Figura 10.5).

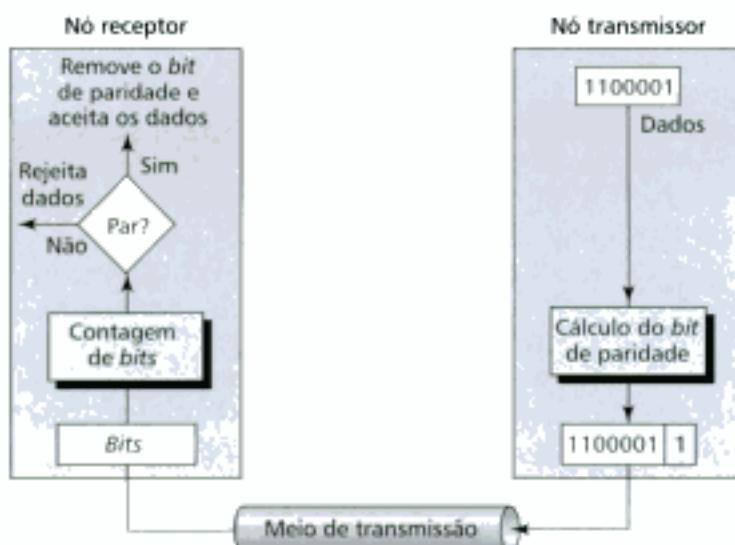


Figura 10.5 Conceito: paridade par.

Somando a quantidade de *bits* em nível 1 do caractere chegamos a 3, ou seja, um número ímpar. Assim, antes de iniciar a transmissão introduzimos o caractere num bloco funcional gerador de paridade. O gerador de paridade conta a quantidade de 1s e adiciona o *bit* de paridade (1 neste caso) no final do caractere. Nesse caso, a quantidade total de *bits* no nível 1 passa a ser 4, ou seja, um número par. O sistema transmite o conjunto (caractere mais o *bit* de paridade) através do meio de transmissão. Quando o conjunto alcança o destino, o receptor passa todos os 8 *bits* por um bloco

* N. de R. T.: A paridade combinada também é conhecida com o nome de paridade bidimensional porque ela organiza os *bits* numa matriz, em linhas e colunas.

funcional verificador de **paridade par**. Se no receptor o conjunto estiver intacto (11000011), ele conta a quantidade de 1s, o que resultará no número 4, e o caractere sem o bit de paridade é aceito como informação válida. Mas, e se a todo o conjunto tiver sido corrompido durante o fluxo? E se, ao invés de 11000011 o receptor receber, por exemplo, 11001011? Nesse caso, quando o bloco verificado de paridade contar os 1s, a quantidade de bits no nível 1 passa a ser 5, ou seja, um número ímpar. Nesse caso, o receptor sabe que ocorreu um erro nos dados em alguma instância do processo e descarta todo o conjunto. Note que por motivo de simplicidade de análise, discutimos o caso onde o bloco verificador de paridade olha apenas a paridade par, isto é, o total dos bits 1 adicionados resulta em um número par. Alguns sistemas podem utilizar a verificação da **paridade ímpar**, isto é, o total dos bits 1 adicionados resulta em um número ímpar. O princípio é o mesmo.

Na verificação da paridade, um bit de paridade é adicionado no final de cada seqüência de dados e o número total de 1s é feito par ou ímpar de acordo com a conveniência.

Exemplo 1

Suponha que a fonte queira enviar a palavra *world*. Em ASCII (veja o Apêndice A), os cinco caracteres são codificados como:

← 1110111 1101111 1110010 1101100 1100100
w o r l d

Cada um dos quatro primeiros caracteres possui um total par de bits em 1. Assim, o bit de paridade é um 0. Entretanto, o último caractere (d) possui três 1s (número ímpar), tal que o bit de paridade é 1, o que torna par a quantidade total 1s nessa seqüência de dados (caractere d mais o bit de paridade). A seqüência abaixo representa a seqüência verdadeira de bits a ser enviada pelo transmissor (os bits de paridade foram colocados sublinhados).

← 11101110 1101111 11100100 1101100 11001001

Exemplo 2

Suponha que a palavra *world* do Exemplo 1 seja recebida sem ser corrompida durante a transmissão.

← 11101110 1101111 11100100 1101100 11001001

O receptor conta os 1s em cada caractere e descobre números pares (6, 6, 4, 4, 4). Os dados são aceitos.

Exemplo 3

Suponha que a palavra *world* do Exemplo 1 seja corrompida durante a transmissão.

← 11111110 1101111 11101100 1101100 11001001

O receptor conta os 1s em cada caractere e descobre números pares e ímpares (7, 6, 5, 4, 4). O receptor sabe que os dados foram corrompidos. Ele os descarta e solicita retransmissão.

Performance

A paridade de caractere pode detectar todos os erros isolados de uma seqüência de dados. Ela também pode detectar rajadas de erros sempre que a quantidade de bits corrompidos for ímpar (1, 3, 5, etc.). Digamos que temos uma seqüência de dados onde o número de 1s, incluindo o bit de paridade, é 6: 1000111011. Se 3 bits quaisquer forem corrompidos, a paridade resultante será ímpar e o erro será detectado. Por exemplo, se a seqüência original for modificada para 111111011 ou 0110111011 ou 1100010011, a paridade passa a ser 9, 7 e 5, respectivamente, todas ímpares. O verificador retornará erros e essas seqüências de dados serão rejeitadas. O mesmo ocorre quando a quantidade de bits corrompidos for um número ímpar qualquer.

Contudo, suponha que apenas 2 bits da seqüência sejam corrompidos na transmissão. Por exemplo, se a seqüência original for modificada para 1110111011 ou 1000111011 ou 1000011010, a paridade passa a ser 8, 6 e 4, respectivamente. Nesses casos, a quantidade de bits 1s nas seqüências de dados continua a exibir paridade par. Embora as seqüências contenham dois bits errados, o

verificador retornará paridade par em cada caso. Significa que esse método não consegue detectar erros onde a quantidade de *bits* corrompidos é par. Se dois *bits* quaisquer sofrerem modificação durante a transmissão, as mudanças nos *bits* irão se cancelar e a seqüência de dados será tida como verdadeira, embora esteja corrompida. O mesmo ocorre quando a quantidade de *bits* corrompidos for um número par qualquer.

A verificação de paridade no nível de caractere detecta somente erros isolados. Ela pode detectar rajadas de erros se, e somente se, o número total de erros em cada seqüência é ímpar.

Verificação da Paridade Combinada

Uma aproximação melhor é a **verificação da paridade combinada**. Neste método, *bits* em bloco são organizados numa tabela (em linhas e colunas). Primeiramente, essa técnica determina o *bit* de paridade de cada seqüência (tipicamente caractere). Então, ela organiza os *bits* numa tabela. Por exemplo, de acordo com a Figura 10.6, se tivermos quatro seqüências de dados, organizadas em quatro linhas e oito colunas, o *bit* de paridade de cada coluna é calculado e adicionado na quinta linha de 8 *bits*. Estes são os *bits* de paridade de todo o bloco. Perceba que o primeiro *bit* de paridade na quinta coluna é calculado com base em todos os *bits* da primeira coluna. O segundo *bit* de paridade é calculado com base em todos os *bits* da segunda coluna e assim por diante. Assim são adicionados 8 *bits* de paridade aos dados originais que são enviados ao receptor.

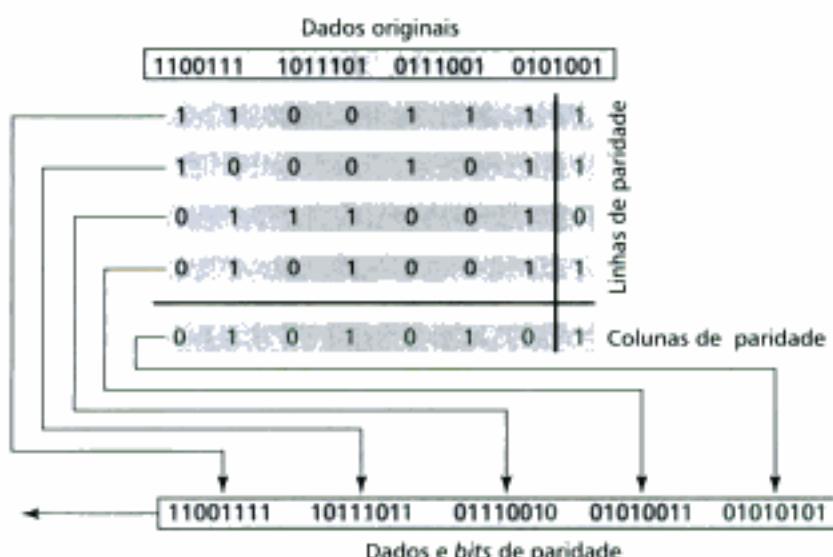


Figura 10.6 Verificação da paridade combinada.

Exemplo 4

Suponha que a fonte transmita o seguinte bloco de dados:

← 10101001 00111001 11011101 11100111 10101010

Suponha ainda que um ruído em rajada de comprimento 8 *bits* interfira no bloco de dados e que esses *bits* sejam corrompidos.

← 10100011 10001001 11011101 11100111 10101010

Quando o receptor verificar os *bits* de paridade, alguns dos *bits* não seguirão a regra de paridade par e todo o bloco é descartado (os *bits* corrompidos aparecem em negrito).

← 10100011 10001001 11011101 11100111 10101010
(bits de paridade)

Na verificação da paridade combinada, os bits são divididos em blocos, formando linhas e colunas, e uma linha redundante de paridade é adicionada ao bloco de dados.

Performance

O teste do bit de paridade combinada melhora as chances de detecção de rajadas de erros. De acordo com o Exemplo 4, n bits de redundância podem detectar facilmente uma rajada de n bits de erros. Uma rajada de erros superior a n bits também é detectada por este método com uma probabilidade muito elevada. Há, porém, um padrão de erros que permanece indefinido quanto à detecção. Se 2 bits de uma certa unidade de dados forem danificados e dois outros exatamente nas mesmas posições da outra unidade da matriz também forem corrompidos, o verificador não irá detectar erros. Considere, por exemplo, dois bytes de dados: 11110000 e 11000011. Se o primeiro e o último bits em cada byte forem corrompidos, modificando os bytes para 01110001 e 01000010, os erros não são detectados por este método.

Cyclic Redundancy Check (CRC)

Outra técnica mais poderosa de verificação de redundância é a **Cyclic Redundancy Check (CRC)***. Diferentemente da verificação de paridade baseada na adição de bits ao bloco de dados, a técnica CRC baseia-se numa divisão binária. Assim, na técnica CRC ao invés do transmissor adicionar bits ao bloco, identificando a paridade desejada, uma seqüência de bits de redundância, denominados bits de CRC, são acrescentados no final do bloco de dados de maneira a tornar todo o bloco resultante divisível por outro número binário predeterminado. No receptor, o bloco de dados é dividido pelo mesmo número binário. O bloco de dados será assumido intacto e aceito se não houver resto da divisão do bloco pelo número binário. Um resto indica que o bloco de dados foi corrompido durante o trânsito, portanto deve ser rejeitado.

Os bits de redundância usados pelo CRC são o resto da divisão do bloco de dados por um divisor binário predeterminado. Para ser válido, um CRC deve possuir duas características: ter exatamente um bit a menos que o divisor e o conteúdo acrescentado no final do bloco de dados deve tornar exato o resultado da divisão do novo bloco pelo divisor, isto é, o resto da divisão deve ser nulo.

Tanto a teoria quanto a aplicação da detecção de erro pela técnica do CRC são diretas. A única complexidade está no cálculo do CRC. Para esclarecermos este processo, iniciaremos com um resumo. Adicionaremos a complexidade necessária à medida que evoluirmos na análise. A Figura 10.7 é um esquema em três passos básicos da técnica CRC.

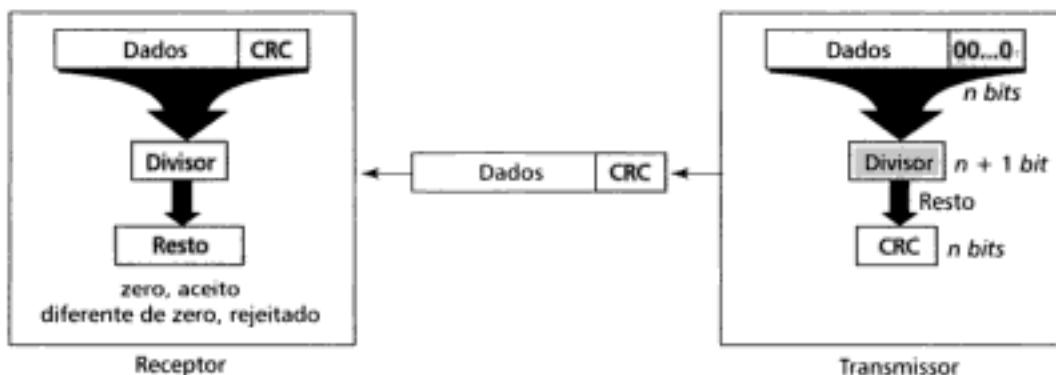


Figura 10.7 O gerador e verificador de CRC.

Primeiro, suponha uma string de dados composta de n zeros (0s) seja acrescentada ao bloco de dados. O número de bits (n) é uma unidade menor que o número de bits predeterminados para o divisor, o qual possui $n + 1$ bits.

Segundo, o novo bloco de dados é dividido pelo divisor, usando um processo denominado divisão binária. O resto da divisão é o CRC.

* N. de R. T.: É comum encontrarmos a expressão teste de redundância ciclica para o acrônimo CRC. Preferimos não traduzi-lo.

Terceiro, o CRC de n bits calculado no segundo passo substitui os Os acrescentados no final do bloco de dados. Note que o CRC pode ser formado apenas de Os.

O bloco de dados chega ao receptor primeiro, seguido do CRC. O receptor verifica toda string como bloco de dados e a divide pelo mesmo divisor utilizado na geração do CRC no lado do transmissor.

Se a *string* for recebida sem erros, o verificador de CRC chega a uma divisão exata, isto é, sem resto, e o bloco de dados é aceito. Se a *string* sofreu alguma modificação no trânsito, a divisão não será exata e o bloco de dados será descartado.

O Gerador de CRC

Um circuito **gerador de CRC** utiliza divisão módulo 2. A Figura 10.8 ilustra este processo. No primeiro passo, o divisor de 4-bits é subtraído dos quatro primeiros bits do dividendo. Cada bit do divisor é subtraído do bit correspondente do dividendo sem que o próximo mais significativo da sequência seja afetado. No exemplo, o divisor (1101) é subtraído dos quatro primeiros bits do dividendo (1001) produzindo 100 (o zero à esquerda foi desconsiderado). O próximo bit do dividendo, não utilizado no passo anterior, é abaixado para igualar o número de bits do resto ao número de bits do divisor. Em seguida, é efetuada a subtração 1000 – 1101. Isso produz 101 e o próximo bit não utilizado do dividendo é abaixado para continuar a divisão. Desse modo, o processo continua até que todos os bits do dividendo sejam utilizados.

Nesse processo, o divisor sempre inicia com um bit 1. O divisor é subtraído da porção do dividendo/resto anterior de igual comprimento (em número de bits). Além disso, o divisor só pode ser subtraído do dividendo/resto cujo bit mais à esquerda valha 1. Quando ocorrer do bit mais à esquerda do dividendo/resto valer 0, uma *string* formada de Os, com o mesmo comprimento do divisor, substitui o divisor nessa etapa do processo. Por exemplo, se o divisor possuir 4-bits, ele é substituído por quatro Os (lembre-se, sempre que estivermos trabalhando com padrões de bits, não com valores quantitativos, 0000 não representa o mesmo que 0). Essa restrição estabelece que, em qualquer passo, a subtração mais à esquerda será sempre 0 – 0 ou 1 – 1, ambas sendo 0. Assim, após a subtração, o bit mais à esquerda do resto será sempre igual a zero, isto é, pode ser desprezado, e o próximo bit do dividendo é abaixado para preencher a quantidade de bits do resto. Perceba que somente o primeiro bit do resto é desprezado – se o segundo bit também vale 0, ele é retido e o dividendo/resto do próximo passo iniciará com 0. Este processo se repetirá até que todos os bits do dividendo tenham sido utilizados.

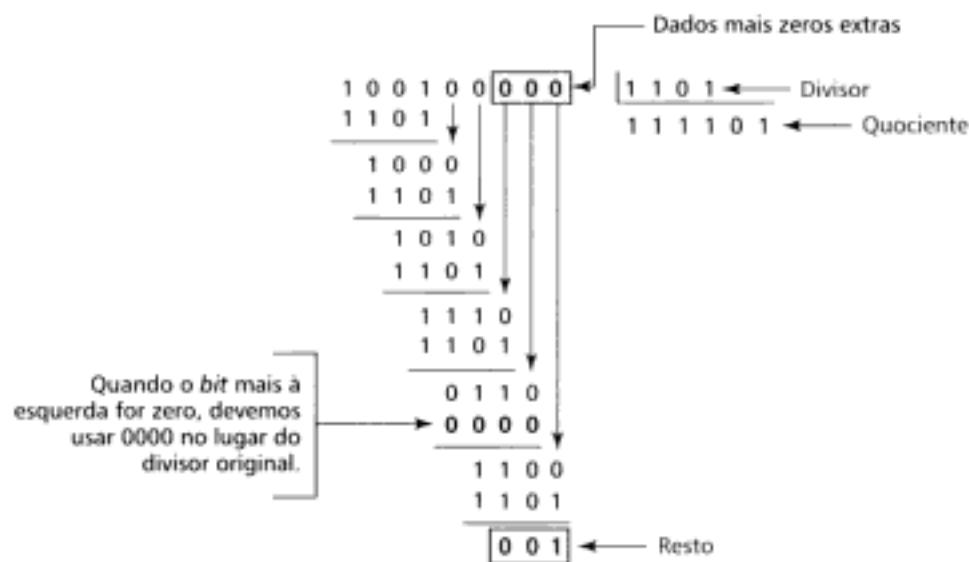


Figura 10.8 Divisão binária no gerador de CRC.

O Verificador de CRC

As funções do **verificador de CRC** são exatamente as mesmas do gerador. Após receber o bloco de dados com o CRC, ele faz a mesma divisão módulo 2. Se o resto é todo formado de 0s, o CRC é retirado do bloco e os dados são aceitos. De outro modo, todo o bloco de dados recebidos é descartado e uma retransmissão de dados é solicitada. A Figura 10.9 ilustra o mesmo processo de divisão no lado do receptor. Na figura, assumimos que não ocorreram erros durante a transmissão. Por isso, o resto é todo formado de 0s e o bloco de dados, sem o CRC, é aceito.

Polinômios

No gerador de CRC, o divisor não é freqüentemente representado como uma *string* de 1s e 0s, mas como um **polinômio gerador** (veja Figura 10.10). A forma polinomial é útil por duas razões: ela é mais curta e pode ser utilizada na demonstração matemática dos conceitos (que foge ao escopo deste livro).

A relação entre as formas binária e polinomial é ilustrada na Figura 10.11.

Um polinômio utilizado no processo de CRC deve possuir no mínimo as seguintes propriedades:

- Não deve ser divisível por x .
- Deve ser divisível pelo binômio $x + 1$.

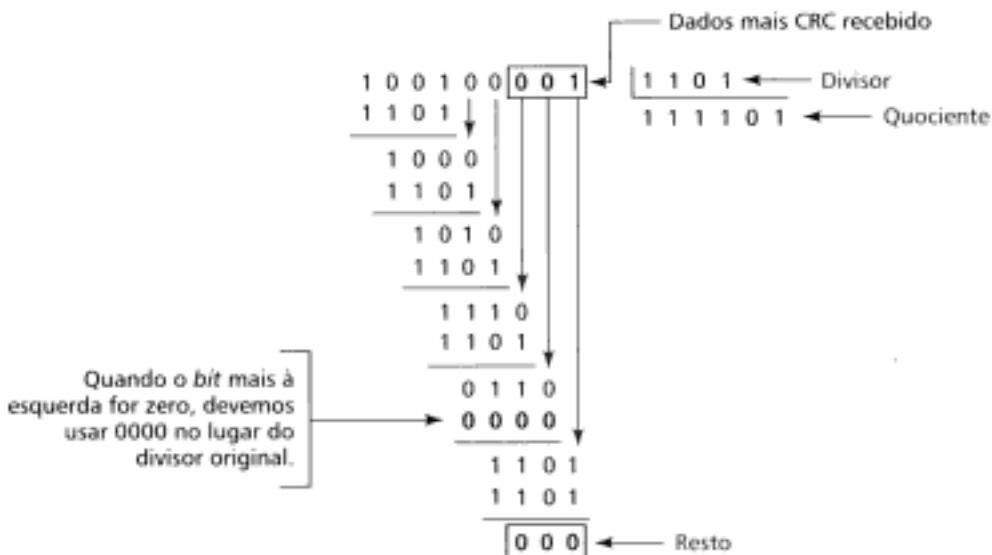


Figura 10.9 Divisão binária no verificador de CRC.

$$x^7 + x^5 + x^2 + x + 1$$

Figura 10.10 Polinômios.

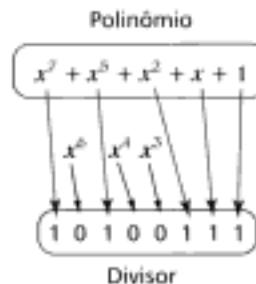


Figura 10.11 Um polinômio representando um divisor.

A primeira condição assegura que serão detectadas todas as rajadas de erros de comprimento igual ao grau do polinômio. A segunda condição assegura que serão detectadas todas as rajadas de erros de comprimento ímpar de *bits* (a prova dessa afirmação foge ao escopo desse livro).

Exemplo 5

Não podemos escolher x (binário 10) ou $x^2 + x$ (binário 110) como polinômio gerador porque ambos são divisíveis por x . Entretanto, podemos escolher $x + 1$ (binário 11) porque não é divisível por x , mas o é por $x + 1$. Podemos também escolher $x^2 + 1$ (binário 101) porque ele é divisível por $x + 1$ (divisão binária).

Polinômios Padronizados

Alguns dos polinômios utilizados pelos protocolos geradores de CRC são mostrados na Tabela 10.1.

TABELA 10.1 Polinômios Padronizados

Nome	Polinômio	Aplicação
CRC-8	$x^8 + x^7 + x + 1$	ATM header
CRC-10	$x^{10} + x^9 + x^5 + x^4 + x^2 + 1$	ATM AAL
ITU-16	$x^{16} + x^{12} + x^5 + 1$	HDLC
ITU-32	$x^{32} + x^{26} + x^{25} + x^{22} + x^{19} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$	LANS

Performance

O método CRC é muito eficiente na detecção de erros. Se o divisor foi escolhido de acordo com as regras mencionadas anteriormente, temos:

1. CRC pode detectar todas as rajadas de erros que afetarem uma quantidade ímpar de *bits*.
2. CRC pode detectar todas as rajadas de erros cujos comprimentos forem menores que ou iguais ao grau do polinômio gerador.
3. CRC pode detectar, com uma probabilidade muito alta, rajadas de erros cujos comprimentos forem maiores que o grau do polinômio gerador.

Exemplo 6

O CRC-12 ($x^{12} + x^{11} + x^5 + x + 1$) detectará todas as rajadas de erros que afetarem uma quantidade ímpar de *bits*, todas as rajadas de comprimento menores que ou iguais a 12 *bits* e, com probabilidade de 99,97%, as rajadas de erros de comprimento superiores a 12 *bits*.

Checksum

O terceiro método de detecção de erros discutido nesse capítulo é denominado **checksum**. Assim como o teste de paridade e CRC, o método de **checksum** baseia-se no conceito de redundância.

Gerador de Checksum

No transmissor, o gerador de **checksum** subdivide o bloco de dados em segmentos iguais de *n-bits* (usualmente $n = 16$). Estes segmentos são adicionados utilizando a aritmética de **complemento de um** (veja Apêndices B e E) de tal modo que a soma (*sum*) também tenha *n bits* de comprimento. Este total (*sum*) é então complementado *bit a bit* e acrescentado no final do bloco de dados original, formando os *bits* de redundância, denominado campo de **checksum**. Assim, todo o bloco de dados estendido é transmitido através da rede. Desse modo, se a soma dos segmentos de dados é *T*, o **checksum** será $-T$ (veja as Figuras 10.12 e 10.13).

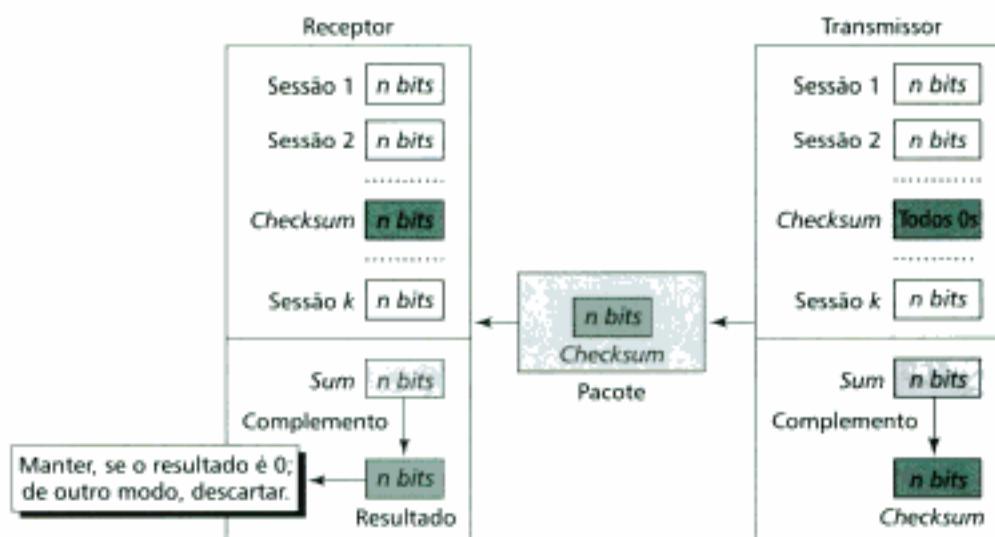


Figura 10.12 Checksum.



Figura 10.13 Unidades de dados e checksum.

O transmissor segue os seguintes passos:

- O bloco de dados é dividido em *k* segmentos de dados, cada qual com *n-bits*.
- Todos os segmentos são somados, através das regras da aritmética de complemento de um, fazendo a soma (*sum*).
- A soma é complementada para gerar o *checksum*.
- O *checksum* é enviado juntamente com os dados.

Verificador de Checksum

O receptor subdivide o bloco de dados exatamente da mesma forma que o transmissor, adiciona todos os segmentos e complementa o resultado. Se os dados estendidos estiverem intactos, o valor total determinado na adição dos segmentos de dados e o campo de *checksum* deverá ser zero. Se o resultado for diferente de zero, o bloco de dados contém pelo menos um erro e o receptor irá descartá-lo (veja Apêndice E).

O receptor segue os seguintes passos:

- O bloco de dados é dividido em *k* segmentos de dados, cada qual com *n-bits*.
- Todos os segmentos são somados, através das regras da aritmética de complemento de um, fazendo a soma (*sum*).
- A soma é complementada para gerar o *checksum*.
- Se o resultado é zero, os dados são aceitos. Senão, eles são rejeitados.

Exemplo 7

Suponha que o seguinte bloco de dados de 16-bits foi enviado usando a técnica de *checksum* de 8-bits.

← 10101001 00111001

Os números são adicionados seguindo a aritmética de complemento de um (veja o Apêndice E).

	10101001
	<u>00111001</u>
<i>Sum</i>	11100010
<i>Checksum</i>	00011101

Assim, o padrão enviado será:

← 10101001 00111001 00011101
Checksum

Exemplo 8

Suponha ainda que os dados enviados no Exemplo 7 cheguem ao receptor livres de erros:

10101001 00111001 00011101

Quando o receptor adicionar os três segmentos, o resultado será todo formado de 1s, o qual sendo complementado resultará em 0s em todos os bits. Assim, não ocorreu erro de transmissão.

	10101001
	<u>00111001</u>
	<u>00011101</u>
<i>Sum</i>	11111111
<i>Complemento</i>	00000000 Significa que a transmissão está OK.

Exemplo 9

Dessa vez, suponha que ocorra uma rajada de erros de comprimento 5 que afete 4-bits de dados.

10101111 11111001 00011101

Quando o receptor adicionar os três segmentos de dados, obterá:

	10101111
	<u>11111001</u>
	<u>00011101</u>
Resultado	1 11000101
Transporte (<i>carry</i>)	1
<i>Sum</i>	11000110
Complemento	00111001 Isto é, o bloco de dados foi corrompido durante a transmissão.

Performance

O método de *checksum* detecta todos os erros envolvendo uma quantidade ímpar de bits, assim como a maioria dos erros envolvendo uma quantidade par. Entretanto, se um ou mais bits de um segmento forem corrompidos e o bit ou bits correspondente(s), ou seja, na mesma posição, do outro segmento também for(em) corrompido(s), a soma (*sum*) da coluna não será modificada e o receptor não detectará o problema. Se o último dígito de um segmento é um 0 e ele for modificado para um 1 durante o trânsito, o último 1 do outro segmento foi modificado para 0, já que não houve indicação de erro. No teste da paridade combinada, dois 0s poderiam sofrer modificação para 1s sem que alterasse a paridade porque a soma não leva os transportes (*carries*) entre colunas em consideração. A técnica de *checksum* retém todos os transportes. Assim, embora dois 0s tenham sido modificados para 1s, não alterando o valor da soma na própria coluna, certamente modificaria o valor da soma na próxima coluna. Mas, o erro será invisível sempre que a inversão de um bit for balanceada por uma inversão oposta no dígito correspondente de outro segmento de dados.

10.3 CORREÇÃO DE ERROS

Os mecanismos de detecção de erros discutidos na seção anterior apontam erro(s), mas não realiza a correção. A **correção de erros** pode ser realizada de muitas formas diferentes. As duas mais comuns são a correção de erros baseada na retransmissão e a correção antecipada de erros.

Correção de Erros por Retransmissão

Na **correção de erros por retransmissão**, quando é detectado um erro o receptor pede ao transmissor que reenvie tudo novamente. Este tipo de correção de erro será discutido em detalhes no Capítulo 11 quando estiverem em pauta os protocolos de controle de fluxo e de erros.

Correção Antecipada de Erros

Na **correção antecipada de erros**, o receptor usa algum código de correção de erros que corrige automaticamente certos tipos de erros. Em teoria é possível corrigir quaisquer tipos de erros automaticamente. Entretanto, os códigos de correção de erros são mais sofisticados que os códigos de detecção de erros e requerem muitos outros *bits* de redundância.

O conceito subjacente à correção de erros pode ser compreendido mais facilmente se examinarmos o caso mais simples: os erros isolados. Como vimos antes, os erros isolados são detectados adicionando um *bit* de redundância (paridade). Um único *bit* adicional consegue detectar erros isolados em qualquer seqüência de *bits* porque ele deve distinguir entre duas condições: erro e sem erro. Já que um *bit* possui dois estados (0 e 1), esses dois estados são suficientes para este tipo de detecção.

Mas, e se, além de detectarmos, quisermos corrigir os erros isolados? Dois estados são suficientes para detectar um erro, mas não para corrigi-lo. Um erro ocorre quando o receptor lê um *bit* 1 como 0 ou vice-versa. Para corrigir o erro, o receptor deve simplesmente inverter o valor do *bit* modificado. Para tanto, antes disso, ele deve saber se o *bit* está errado ou não. O segredo da detecção de erro é, assim, localizar o(s) *bit*(s) inválido(s).

Por exemplo, para corrigir um erro isolado num caractere ASCII, o código de correção de erro deve determinar qual dos 7-*bits* foi modificado antes de modificá-lo. Nesse caso, o receptor tem que distinguir entre oito estados diferentes: sem erro, erro na posição 1, erro na posição 2, ..., erro na posição 7. Isso requer *bits* de redundância suficientes para mostrar todos os oito estados.

À primeira vista parece que um código de redundância de 3-*bits* seria suficiente porque 3-*bits* podem representar oito estados diferentes (000 a 111) e, assim, indicar as localizações das oito posições diferentes. Mas, o que ocorre se o erro ocorrer nos próprios *bits* de redundância? Sete *bits* de dados (o caractere ASCII) mais 3-*bits* de redundância perfazem 10-*bits*. Entretanto, três *bits* só mapem apenas oito possibilidades. Assim, *bits* adicionais são necessários para cobrir todas as possibilidades de localização de erros.

Para calcular o número de *bits* de redundância (*r*) requeridos na correção de *m*-*bits* de dados deve ser estabelecida uma relação entre *m* e *r*. Com *m*-*bits* de dados e *r*-*bits* de redundância adicionados, o comprimento do código resultante é *m* + *r*.

Se o número total de *bits* de dados do bloco for *m* + *r*, então *r* deve ser capaz de identificar no mínimo *m* + *r* + 1 estados diferentes. Destes, um estado é utilizado para condição sem erro e *m* + *r* estados indicam a localização de um erro em cada uma das *m* + *r* posições.

Então, *m* + *r* + 1 estados serão determinados por *r*-*bits*, que por si conseguem identificar 2^r estados diferentes. Logo, 2^r deve ser maior do que ou igual a *m* + *r* + 1:

$$2^r \geq m + r + 1$$

O valor de *r* pode ser determinado substituindo o valor de *m* (o comprimento original de dados a ser transmitidos) na desigualdade. Por exemplo, se *m* = 7 (como num código ASCII de 7-*bits*), o menor valor de *r* que satisfaz a desigualdade é 4:

$$2^4 \geq 7 + 4 + 1$$

A Tabela 10.2 mostra alguns dos possíveis valores de m e os valores correspondentes de r .

TABELA 10.2 Relação entre dados e bits de redundância

Número de bits de dados m	Número de bits de redundância r	Total de bits $m + r$
1	2	3
2	3	5
3	3	6
4	3	7
5	4	9
6	4	10
7	4	11

Código Hamming

Hamming propôs uma solução prática. O **código Hamming** pode ser aplicado em unidades de dados de qualquer tamanho e utiliza a relação entre os bits de dados e de redundância discutidos acima. Por exemplo, o código ASCII de 7-bits requer 4-bits de redundância que podem ser adicionados no final do código ou intercalar os bits de dados originais. Na Figura 10.14, esses bits foram colocados nas posições 1, 2, 4 e 8. Para exemplificar, vamos nos referir aos bits nas posições r_1 , r_2 , r_4 e r_8 .

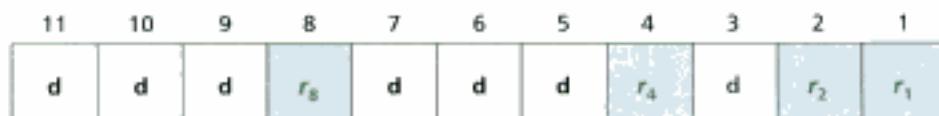


Figura 10.14 Posições dos bits de redundância no código Hamming.

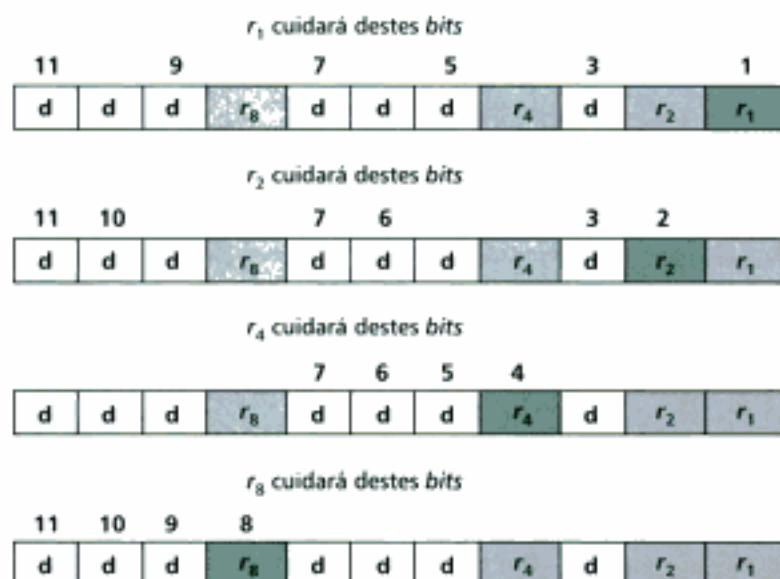
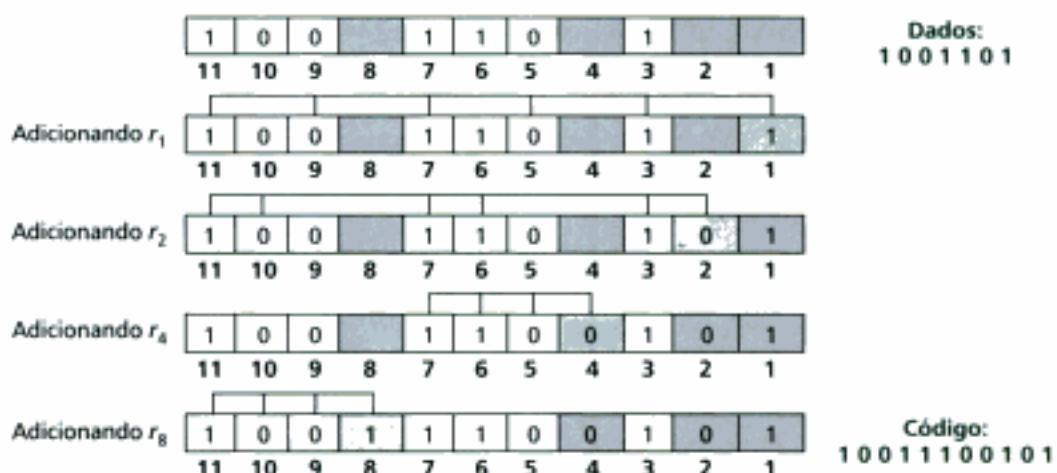
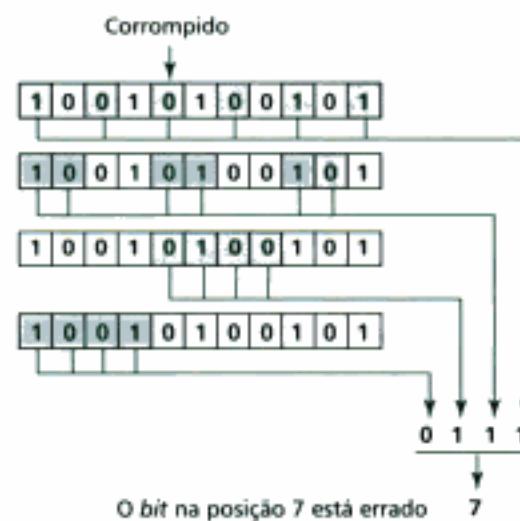
No código Hamming, cada r -bit é o bit de paridade para uma combinação de bits de dados, conforme mostrado abaixo:

- r_1 : bits 1, 3, 5, 7, 9, 11
- r_2 : bits 2, 3, 6, 7, 10, 11
- r_4 : bits 4, 5, 6, 7
- r_8 : bits 8, 9, 10, 11

Cada bit de dados pode ser incluído em mais de um cálculo. Por exemplo, nas seqüências acima, cada um dos bits de dados originais aparece em pelo menos dois conjuntos, enquanto os r -bits estão incluídos em somente um (veja a Figura 10.15).

Determinando os Valores de R A Figura 10.16 ilustra uma implementação do código Hamming para um caractere ASCII. No primeiro passo, colocamos cada bit do caractere original na posição apropriada dentro da seqüência de 11-bits. Nos passos subsequentes, determinamos paridades par para várias combinações de bits. O valor da paridade para cada combinação é o valor do r -bit correspondente.

Detecção e Correção de Erro Agora, imagine que durante a transmissão acima, o bit na posição 7 foi modificado de 1 para 0. O receptor pega a transmissão e recalcula os 4 novos bits de paridade usando os mesmos bits utilizados pelo transmissor mais o r -bit de paridade relevante para cada conjunto (veja a Figura 10.17). Em seguida, o receptor monta os valores das novas paridades no número binário na ordem indicada pela posição r (r_8 , r_4 , r_2 , r_1). No exemplo, este passo resulta no número binário 0111 (7 em decimal), o qual revela a localização exata do bit corrompido.

**Figura 10.15** Cálculo dos bits de redundância.**Figura 10.16** Exemplo de cálculo do bit de redundância.**Figura 10.17** Detecção de erros usando o código Hamming.

Uma vez que o *bit* foi identificado, o receptor pode inverter o valor do *bit* e corrigir o erro. A beleza dessa técnica é que ela pode ser implementada facilmente do ponto de vista de *hardware* e o código é corrigido antes mesmo do receptor tomar conhecimento dele.

Correção da Rajada de Erros

Embora o código de Hamming não possa corrigir uma rajada de erro diretamente, é possível rearranjar os dados e, então, aplicar o código. Em vez de enviar todos os *bits* juntos num bloco de dados, podemos organizar N unidades em colunas e, então, enviar o primeiro *bit* de cada bloco. Desse modo, se ocorrer uma rajada de erros de M -*bits* ($M < N$), então o erro não modificará os M -*bits* de uma mesma unidade, mas apenas um único *bit* de cada unidade. Utilizando o esquema de Hamming, podemos corrigir o *bit* corrompido em cada unidade. A Figura 10.18 ilustra um exemplo.

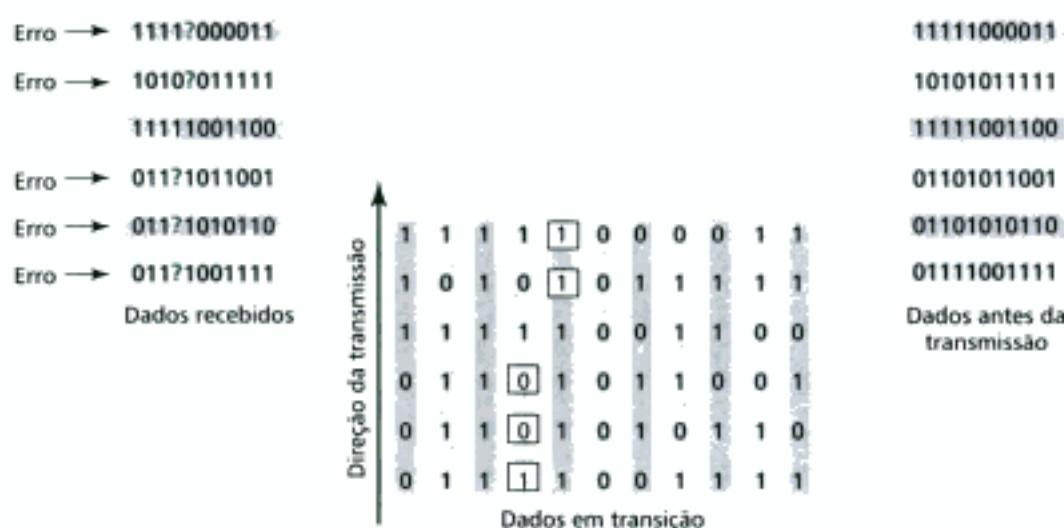


Figura 10.18 Exemplo de correção de uma rajada de erros.

Na Figura 10.18, são enviadas seis unidades de dados, cada qual contendo um caractere com os *bits* de redundância (o código Hamming). Organizamos os *bits* em linhas e colunas. Então, enviamos a primeira coluna, em seguida a segunda e assim por diante. Os *bits* corrompidos pela rajada de erros aparecerem destacados nos retângulos. Cinco *bits* consecutivos foram corrompidos durante a transmissão. Assim, quando esses *bits* chegarem ao destino e serem reorganizados novamente em unidades de dados, cada *bit* corrompido pertence a uma unidade diferente e será automaticamente corrigido. O truque aqui é permitir que a rajada corrompa apenas um *bit* de cada unidade de dados.

10.4 TERMOS-CHAVE

<i>Bit</i> de paridade	Erro isolado
<i>Checksum</i>	Gerador de CRC
Código de Hamming	Paridade ímpar
Complemento de um	Paridade par
Correção antecipada de erros	Polinômio gerador
Correção de erros	Rajada de erros
Correção de erros por retransmissão	Redundância
Cyclic Redundancy Check (CRC)	Teste de paridade
Detecção de erros	Verificação da paridade combinada
Erro	Verificador de CRC

10.5 RESUMO

- Os erros podem ser classificados como erros isolados ou rajada de erros. Num erro isolado, um único *bit* de erro é encontrado na unidade de dados (*byte*, caractere, pacote, etc.). Uma rajada de erros possui dois ou mais erros por unidade de dados.
- Redundância é o nome que se dá aos *bits* extra utilizados na detecção de erros.
- Existem três métodos bastante difundidos de redundância: teste de paridade, CRC e o *checksum*.
- No teste baseado na paridade, um único *bit* extra (*bit* de paridade) é necessário para a detecção de erros.
- O teste baseado na paridade pode detectar somente uma quantidade ímpar de erros. Ela não pode detectar uma quantidade par de erros na unidade de dados.
- Na paridade combinada, uma unidade de dados redundante segue uma unidade de *n-bits* de dados.
- A poderosa técnica de CRC acrescenta uma sequência de *bits* redundantes gerados a partir da divisão binária da unidade de dados.
- Frequentemente, no gerador de CRC, o divisor é representado por um polinômio gerador.
- Erros podem ser corrigidos através da retransmissão ou da antecipação.
- O código Hamming é um método de correção de erros que utiliza *bits* de redundância. A quantidade de *bits* utilizados é uma função do tamanho dos *bits* de dados.
- No código Hamming, dados *m-bits* de dados, use a fórmula $2^r \geq m + r + 1$ para determinar *r* (a quantidade necessária de *bits* de redundância).
- Modificando a ordem de transmissão dos *bits* de uma unidade de dados, o código Hamming pode ser utilizado na correção de rajadas de erros.

10.6 PRATIQUE OS CONHECIMENTOS ADQUIRIDOS

Questões de Revisão

1. Como um erro isolado difere de uma rajada de erros?
2. Discuta o conceito de redundância na detecção de erros.
3. Quais os três tipos de teste de redundância utilizados na comunicação de dados?
4. Como o *bit* de paridade pode detectar erro numa unidade de dados?
5. Qual é a diferença entre as paridades par e ímpar?
6. Discuta o teste de paridade e os tipos de erros que ele detecta ou não.
7. Como o teste de paridade se relaciona à verificação da paridade combinada?
8. Discuta o teste de paridade combinada e os tipos de erros que ele detecta ou não.
9. Que tipo de objeto um gerador de CRC acrescenta numa unidade de dados?
10. Qual é a relação entre o tamanho do CRC (resto) e o divisor?
11. No receptor, como o verificador de CRC sabe que a unidade de dados recebida foi corrompida?
12. Quais são as condições aplicáveis ao polinômio no gerador de CRC?
13. De que forma a técnica de CRC é superior à paridade combinada?
14. Qual é o método de detecção de erros utilizado pelos protocolos de camada superior?
15. Que tipo de aritmética é utilizada na adição dos segmentos de dados no gerador e no verificador de *checksum*?
16. Liste os passos necessários à determinação do *checksum*.
17. Como o verificador de *checksum* sabe que os dados recebidos foram corrompidos?
18. Que tipo de erros não pode ser detectado pela técnica *checksum*?
19. Qual é a fórmula para calcular a quantidade de *bits* de redundância requerida para corrigir um *bit* errado numa certa unidade de dados?
20. Qual é o propósito do código Hamming?
21. Como o código Hamming pode ser utilizado na correção de rajadas de erros?

Questões de Múltipla Escolha

22. Que método de detecção de erros olha para o *bit* de paridade de cada unidade tanto quanto para toda a unidade de dados (incluindo os *bits* de paridade)?
- Teste do *bit* de paridade
 - Teste da paridade combinada
 - CRC
 - Checksum*
23. Que método de detecção de erros utiliza a aritmética de complemento de um?
- Teste do *bit* de paridade
 - Teste da paridade combinada
 - CRC
 - Checksum*
24. Que método de detecção de erros consiste de um único *bit* redundante por unidade de dados?
- Teste do *bit* de paridade
 - Teste da paridade combinada
 - CRC
 - Checksum*
25. Que método de detecção de erros utiliza polinômios geradores?
- Teste do *bit* de paridade
 - Teste da paridade combinada
 - CRC
 - Checksum*
26. Qual das opções abaixo melhor define um erro isolado?
- Um único *bit* é invertido
 - Um único *bit* é invertido por unidade de dados
 - Um único *bit* é invertido por transmissão
 - Qualquer uma das opções anteriores
27. Se o caractere G, codificado em ASCII, é enviado e o caractere D é recebido, que tipo de erro ocorreu durante a transmissão?
- Erro isolado
 - Erros múltiplos
 - Rajada
 - Recuperável
28. Se o caractere H, codificado em ASCII, é enviado e o caractere I é recebido, que tipo de erro ocorreu durante a transmissão?
- Erro isolado
 - Erros múltiplos
 - Rajada
 - Recuperável
29. No teste de redundância cíclica, quem é o CRC?
- O divisor
 - O quociente
 - O dividendo
 - O resto
30. No teste de redundância cíclica, o divisor tem _____ o CRC.
- O mesmo tamanho que
 - 1 *bit* a menos que
 - 1 *bit* a mais que
 - 2 *bits* a mais que
31. Se a unidade de dados é a seqüência 1111111, o divisor 1010 e o resto 110, que opção abaixo representa o dividendo utilizado pelo receptor?
- 111111011
 - 111111110
 - 1010110
 - 110111111
32. Se a unidade de dados é 111111 e o divisor 1010, qual é o dividendo utilizado pelo transmissor?
- 111111000
 - 1111110000
 - 111111
 - 1111111010
33. Se a paridade ímpar for escolhida na detecção de erros em ASCII, o número de 0s por símbolo de 8-bits é _____.
- Par
 - Ímpar
 - Indeterminado
 - 42
34. A soma do *checksum* e dos dados no receptor é _____, se não incidir erros durante a transmissão.
- 0
 - +0
 - O complemento do *checksum*
 - O complemento dos dados
35. O código Hamming é um método de _____.
- Detecção de erros
 - Correção de erros
 - Encapsulamento de erros
 - (a) e (b)
36. No CRC, não haverá erros se o resto no receptor for _____.
- Igual ao resto do transmissor
 - Zero
 - Diferente de zero
 - O quociente do transmissor

37. No CRC, o quociente do transmissor
- Torna-se o dividendo no receptor
 - Torna-se o divisor no receptor
 - É descartado
 - É o resto
38. Que método de detecção de erros utiliza *bits* de paridade?
- Teste do *bit* de paridade
 - Teste da paridade combinada
 - CRC
 - (a) e (b)
39. Que método de detecção de erros pode detectar um erro isolado?
- Teste do *bit* de paridade
 - Teste da paridade combinada
 - CRC
 - Todas acima
40. Que método de detecção de erros detecta uma rajada de erros?
- Teste do *bit* de paridade
 - Teste da paridade combinada
41. No gerador de CRC, _____ é adicionado(a) à unidade de dados antes do processo de divisão.
- Uma *string* de 0s
 - Uma *string* de 1s
 - Um polinômio
 - Um resto CRC
42. No gerador de CRC, _____ é adicionado(a) à unidade de dados após o processo de divisão.
- Uma *string* de 0s
 - Uma *string* de 1s
 - Um polinômio
 - Um resto CRC
43. No verificador de CRC, _____ significa que a unidade de dados foi corrompida.
- Uma *string* de 0s
 - Uma *string* de 1s
 - Uma *string* alternada de 0s e 1s
 - Um resto diferente de zero

Exercícios

44. Qual é o efeito máximo de uma rajada de um ruído de 2ms de duração numa transmissão de dados a:
- 1500bps?
 - 12.000bps?
 - 96.000bps?
45. Assumindo paridade par, determine o *bit* de paridade para cada uma das seguintes unidades de dados.
- 1001011
 - 0001100
 - 1000000
 - 1110111
46. A unidade 01101011 é recebida pelo receptor. Se o sistema estiver utilizando paridade par, a unidade está certa ou errada? Justifique.
47. Um sistema utiliza paridade combinada. Determine a paridade da unidade para as duas unidades de dados a seguir. Assuma paridade par.
10011001 01101111
48. Dada a seqüência de 10-*bits* 1010011110 e o divisor 1011, determine o CRC. Teste sua resposta.
49. Dado o resto 111, a unidade de dados 10110011 e o divisor 1001, existe algum erro nessa unidade?
← 0011101 1100111 1111111
0000000
50. Calcule o *checksum* para a seguinte seqüência de *bits*. Assuma um segmento de 16-*bits* de tamanho.
1001001110010011
1001100001001101
51. Determine o complemento de um de 1110010001110011.
52. Adicione 1110001 e 00011100 utilizando complemento de um. Interprete o resultado.
53. Para cada um dos seguintes tamanhos da unidade de dados, determine a quantidade mínima de *bits* de redundância necessários na correção de um erro isolado.
- 12
 - 16
 - 24
 - 64
54. Construa o código Hamming para a seqüência 10011101.
55. Determine os *bits* de paridade para os seguintes padrões usando paridade simples. Repita o exercício utilizando paridade combinada. Assuma paridade par.

56. Um transmissor envia 01110001 e um receptor recebe 01000001. Se for utilizado teste de paridade, o receptor consegue detectar o erro?

57. O bloco de dados a seguir é recebido por um sistema utilizando paridade combinada par. Quais *bits* estão errados?

← 1 0 0 1 0 1 0 1 0 1 0 0 1 1 1 1
1 1 0 1 0 0 0 0 1 1 0 1 1 0 1 1

58. Um sistema utilizando paridade combinada envia um bloco de 8 *bytes*. Quantos *bits* de redundância são necessários por bloco? Qual é a razão entre os *bits* úteis (informação original) e a quantidade total de *bits*?

59. Se um divisor é 101101, quantos *bits* tem o CRC?

60. Determine o equivalente binário de $x^8 + x^3 + x + 1$.

61. Determine o equivalente polinomial de 100001110001.

62. A seqüência 11001100111 chega ao receptor. Se o receptor utiliza o algoritmo de codificação Hamming o resultado é 0101. Que *bit* está errado? Qual é o código correto?

63. Na correção de erros isolados, um código de 3-*bits* pode estar num dos quatro estados: sem erro, primeiro *bit* errado, segundo *bit* errado e terceiro *bit* errado. Quantos dos 3-*bits* seriam redundantes para corrigir este código? Quantos *bits* formam os dados verdadeiros?

64. Usando a lógica do Exercício 63, determine quantos *bits* de redundância devem estar presentes num código de 10-*bits* para detectar um erro.

65. O código 11110101101 foi recebido. Usando o algoritmo de codificação Hamming, qual é o código enviado originalmente?

Controle do Enlace de Dados e Protocolos

Para que haja comunicação de dados são necessários dois dispositivos trabalhando juntos, um enviando e outro recebendo. Até mesmo o sistema de comunicação mais básico requer uma grande quantidade de esforços entre as partes para que a troca de dados ocorra de modo inteligível. As funcionalidades mais relevantes da camada de enlace de dados são promover o **controle de fluxo** e o **controle de erros**. Coletivamente, essas funcionalidades são conhecidas como **controle do enlace de dados**.

Neste capítulo, logo de partida, definiremos informalmente o controle de fluxo e de erros. Em seguida, introduziremos três mecanismos que permitem o controle de fluxo e de erros. Finalmente, discutiremos um protocolo da camada de enlace bastante conhecido: o HDLC.

11.1 CONTROLE DE FLUXO E CONTROLE DE ERRO

O controle de fluxo e o controle de erro são as funções principais da camada de enlace. Vamos defini-los informalmente.

Controle de Fluxo

O controle de fluxo coordena o volume de dados que podem ser enviados antes de receber um *ack* (abreviação de *acknowledgment*, que significa confirmação ou reconhecimento) e é uma das responsabilidades mais importantes da camada de enlace. Em muitos protocolos, o controle de fluxo é um conjunto de procedimentos que informa ao transmissor a quantidade de dados que ele pode transmitir antes que um *ack* seja recebido do dispositivo receptor. O fluxo de dados não deve permitir que o dispositivo receptor seja inundado pelo transmissor. Todo dispositivo receptor possui um limite de velocidade, para o qual o fluxo de dados de entrada pode ser processado, e uma quantidade de memória onde os dados de entrada são armazenados. O receptor deve ser dotado da capacidade de informar ao transmissor que o limite de capacidade está próximo de ser alcançado e requer uma taxa de transmissão menor (com menos *frames* ou até mesmo a parada completa, mas temporária, da transmissão). O receptor deve verificar e processar os dados de entrada antes que eles sejam utilizados. Freqüentemente, este tipo de processamento ocorre em velocidades muito inferiores à taxa de transmissão dos dados. Por essa razão, os dispositivos receptores são equipados com um segmento de memória, denominado *buffer*, reservado ao armazenamento de dados recém-chegados até que eles possam ser processados. Se o espaço em *buffer* começar a ficar comprometido,

do, o receptor deve ser capaz de comunicar-se com o transmissor e solicitar uma parada de transmissão até que ele possa receber dados novamente.

O controle de fluxo refere-se a um conjunto de procedimentos utilizados para restringir o volume de dados que o transmissor pode enviar sem esperar por um ack.

Controle de Erros

Controle de erros é tanto uma técnica de detecção quanto de correção de erros. Ele permite ao receptor informar ao transmissor sobre quaisquer *frames* (quadros) perdidos ou corrompidos numa transmissão, coordenando a retransmissão dos *frames*, realizadas pelo transmissor, que porventura tenham sido rejeitados. Na camada de enlace, o termo *controle de erros* refere-se primeiramente ao método de detecção e retransmissão. O controle de erro implementado na camada de enlace é freqüentemente o mais simples: toda vez que um erro é detectado num processo de transmissão, os *frames* especificados são retransmitidos. Este processo é denominado **Automatic Repeat Request (ARQ)**.

O controle de erros implementado na camada de enlace é baseado em ARQ (Automatic Repeat Request) que são protocolos dedicados à retransmissão de dados.

Mecanismos de Controle de Fluxo e Erros

Nesta seção, introduziremos três mecanismos de controle de fluxo e de controle de erros*: Stop-and-Wait ARQ, Go-Back-N ARQ e Selective-Repeat ARQ. Embora, às vezes, estes três sejam referidos como protocolos, preferimos utilizar o termo mecanismos.

11.2 STOP-AND-WAIT ARQ

O mecanismo **Stop-and-Wait ARQ** é o procedimento mais simples de controle de fluxo e erros. Possui as seguintes características:

- O dispositivo transmissor mantém uma cópia do último *frame* transmitido até receber uma resposta de confirmação para este *frame*. Manter uma cópia possibilita ao transmissor retransmitir *frames* perdidos ou corrompidos que porventura o receptor solicite.
- Para identificação, tanto o *frame* de dados quanto o *frame* de **acknowledgment (ACK)** são numerados alternadamente como 0 e 1. Um *frame* de dados 0 é confirmado por um *frame* ACK 1 de resposta, indicando que o receptor aceitou o *frame* de dados 0 e espera o *frame* de dados 1. Esta forma de identificação permite enumerar os *frames* de dados no caso de transmissão duplicada (importante nos casos de perdas ou atrasos de ACKs, como veremos adiante).
- Um *frame* perdido ou corrompido é tratado da mesma maneira pelo receptor. Caso receptor detecte erro(s) no *frame* recebido, ele simplesmente o descarta e não envia uma resposta de ACK. Se o receptor receber *frames* fora de ordem (0 ao invés de 1 ou vice-versa), ele sabe que um *frame* foi perdido. Assim, o receptor descarta o *frame* recebido fora de ordem.
- O transmissor possui uma variável de controle, a qual denotaremos *S*, que sustenta o número do *frame* recentemente enviado (0 ou 1). O receptor também possui uma variável de

* N. de R. T.: O nome dos mecanismos ou procedimentos de controle de fluxo e de erros Stop-and-Wait ARQ, Go-Back-N ARQ e Selective-Repeat ARQ foram mantidos em inglês por serem termos consagrados no jargão da comunicação de dados. Entretanto, aqueles que preferirem podem utilizar as seguintes versões em português: algoritmo de bit alternado, janela *n* com retransmissão integral e janela *n* com retransmissão seletiva, respectivamente.

controle, a qual denotaremos R , que sustenta o número do próximo *frame* que o receptor espera receber (0 ou 1).

- O transmissor dispara um relógio no exato instante que envia um *frame*. Se uma resposta ACK não for recebida dentro de um intervalo de tempo predefinido, o transmissor assume que houve uma perda ou dano desse *frame* e o reenvia.
- O receptor envia respostas positivas (ACKs) somente para *frames* recebidos e aceitos. Se um *frame* for rejeitado, ele não comunica ao transmissor pedindo retransmissão (é comum dizer que a linha é mantida em silêncio para os *frames* perdidos ou corrompidos). O número da confirmação sempre define o número do próximo *frame* esperado. Se o *frame* 0 é recebido, o ACK 1 é enviado. Logo, se o *frame* 1 é recebido, ACK 0 é enviado.

Operação

Durante a transmissão de um *frame*, podem ocorrer quatro situações diferentes: operação normal, *frame* perdido, ACK perdido e retardo de ACK.

Operação Normal

Numa transmissão normal, o transmissor envia o *frame* 0 e espera receber o ACK 1. Quando o ACK 1 é recebido, ele envia o *frame* 1 e aguarda pela chegada de um ACK 0 e assim por diante. O ACK deve ser recebido antes que expire o relógio de cada *frame*. A Figura 11.1 ilustra *frames* de transmissões bem-sucedidas.

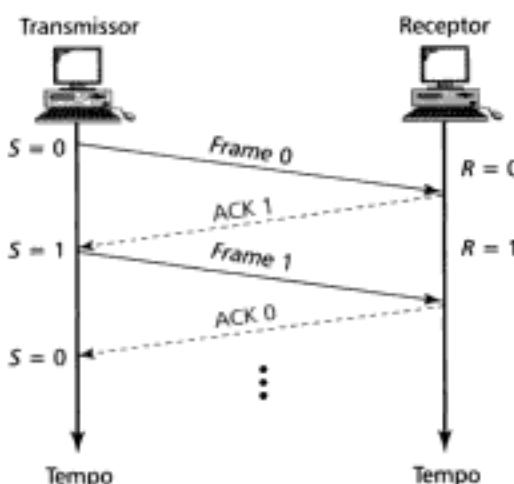


Figura 11.1 Operação normal.

Frame Perdido ou Corrompido

Nesta situação, o *frame* é tratado da mesma forma pelo receptor: quando chega um *frame* corrompido no receptor, ele o descarta. Isso é essencialmente o mesmo que um *frame* perdido do ponto de vista do transmissor. O receptor permanece em silêncio em relação ao *frame* e mantém o valor atual da variável R . Por exemplo, na Figura 11.2, o transmissor envia o *frame* 1, mas ele é perdido. O receptor nada faz e o valor de R (1) é mantido. É enviada uma cópia do *frame* 1 após expirar o relógio do transmissor.

ACK Perdido

Uma resposta ACK perdida ou corrompida é tratada da mesma forma pelo transmissor: se o transmissor receber um ACK corrompido, ele o descarta. A Figura 11.3 mostra a perda de um ACK 0. dessa forma, o transmissor não tem como saber que o *frame* 1 foi recebido. Logo, o transmissor retransmite o *frame* 1 assim que expirar o relógio desse *frame*. Note que o receptor já recebeu o *frame* 1 e espera receber o *frame* 0 ($R = 0$). Assim, ele descarta silenciosamente a cópia do *frame* 1.

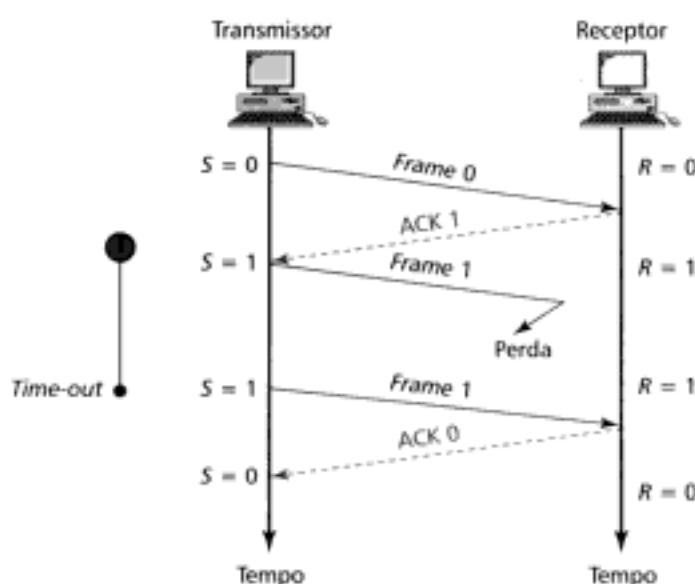


Figura 11.2 Stop-and-Wait ARQ: perda do frame.

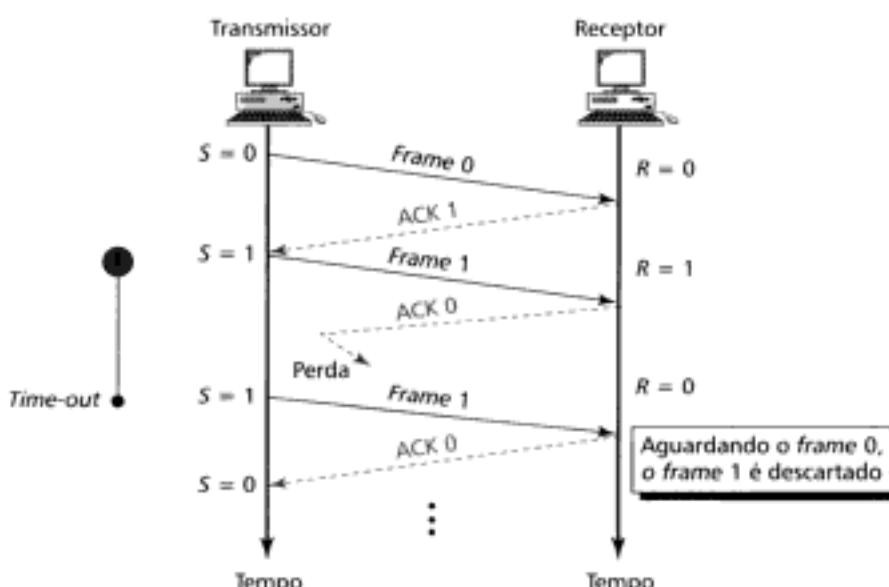


Figura 11.3 Stop-and-Wait ARQ: perda do frame ACK.

O estudante atencioso percebeu através deste exemplo a importância da numeração dos frames. Se os frames não fossem numerados, o receptor trataria o frame 1 como um frame novo, não cópia (duplicata).

No mecanismo Stop-and-wait ARQ, a numeração dos frames evita que o receptor mantenha cópia (duplicata) de frames.

Atraso de ACK (Delayed ACK)

Outro problema previsível é um atraso de ACK. O receptor pode atrasar a resposta de um ACK ou, por algum problema com o link, a resposta pode não chegar ao transmissor. A Figura 11.4 mostra o atraso do ACK 1. Ele é recebido após o relógio do frame 0 ter expirado. Nessas alturas, o transmissor já retransmitiu uma cópia do frame 0. Entretanto, o valor de R no receptor ainda é 1, o qual indica que o receptor espera receber o frame 1. Assim, a única medida adotada pelo receptor é desconsiderar a duplicata do frame 0.

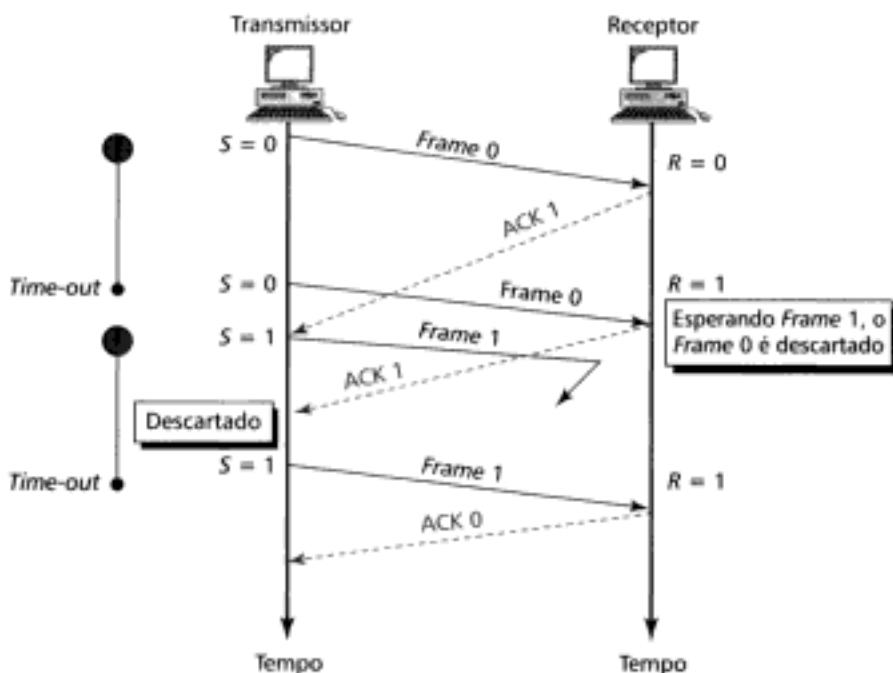


Figura 11.4 Stop-and-Wait ARQ: atraso do ACK.

O transmissor tem que receber dois ACKs, aquele atrasado e outro correspondente ao recebimento da duplicata (*frame 0*). O segundo ACK 1 é descartado.

Vamos examinar a Figura 11.4 novamente para compreendermos porque precisamos de ACK numerados. O *frame 1* é enviado após ACK 1 (atrasado) ser recebido pelo transmissor. Contudo, o *frame 1* é perdido e nunca alcança o receptor. Então, o transmissor recebe a resposta correspondente ao envio do ACK 1 do *frame* duplicado. Se os ACKs não fossem numerados, o transmissor interpretaria o segundo ACK como a confirmação do *frame 1*. A numeração das respostas ACKs provê um método para manter um registro de recebimento dos *frames* de dados.

A numeração de ACKs cuida do problema do atraso de ACKs seguido da perda do próximo *frame*.

Transmissão Bidirecional

O mecanismo de *Stop-and-wait* discutido é essencialmente unidirecional. Porém, como uma transmissão também pode ser bidirecional, se existirem dois canais separados para transmissão em modo *full-duplex* ou compartilhamento do canal em modo *half-duplex*, cada um dos componentes do sistema de comunicação de dados (transmissor e receptor) necessita tanto da variável S quanto da variável R para controlar, nas duas pontas, os *frames* enviados e esperados.

Piggybacking

O **piggybacking (superposição)** é um método que combina (superpõe) um *frame* de dados a um ACK. Por exemplo, na Figura 11.5, as estações A e B têm dados a transmitir. Em vez de enviar dados e frames de ACKs separadamente, a estação A envia um *frame* de dados que inclui um ACK. A estação B comporta-se de maneira similar.

O método de *piggybacking* pode preservar banda do canal porque o *overhead* (sinalização) do *frame* de dados e do *frame* de ACK (endereços, CRC, etc.) pode ser combinado em um único *frame*.

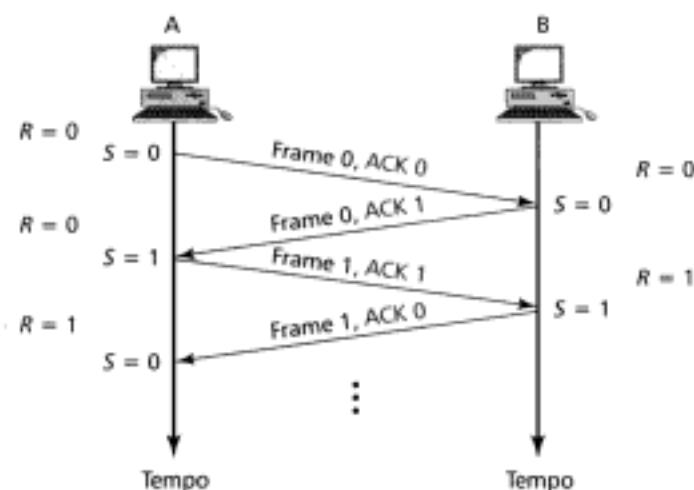


Figura 11.5 Piggybacking.

11.3 GO-BACK-N ARQ

No mecanismo Stop-and-Wait ARQ, em qualquer instante de tempo, um único *frame* é enviado pelo transmissor, que esperada a confirmação ACK. Do ponto de vista de utilização do meio de transmissão esta não é uma boa solução. Para melhorar a eficiência da comunicação, seria interessante enviar múltiplos *frames* enquanto o transmissor espera pelo ACK. Noutras palavras, necessitariamos deixar mais de um *frame* pendente. Dois protocolos utilizam este conceito: **Go-Back-N ARQ** e **Selective Repeat ARQ**. Examinaremos o primeiro nessa seção e deixaremos o segundo para a Seção 11.4.

O protocolo Go-Back-N ARQ prevê o envio de W *frames* antes que a resposta do primeiro ACK seja recebida pelo transmissor. O transmissor mantém uma cópia de todos os *frames* enviados até que o respectivo ACK retorne. Este procedimento é mais complexo que o Stop-and-Wait ARQ, por isso, leva muitas outras características adicionais em consideração.

Seqüência de Números

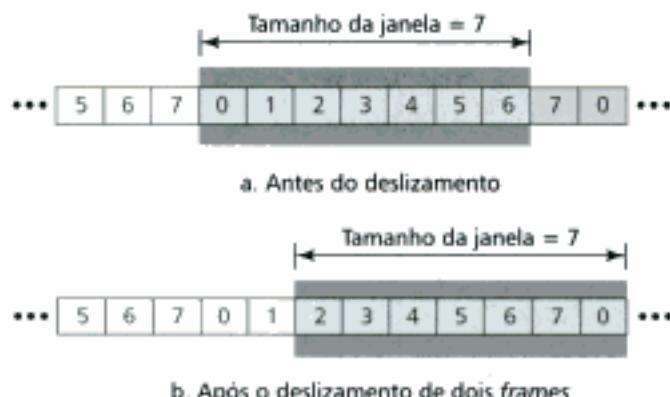
Os *frames* da estação transmissora são numerados seqüencialmente. Contudo, é necessário incluir no cabeçalho o número de seqüência de cada *frame* (*header*), além de estabelecermos um limite. Se o cabeçalho de um *frame* permite m -bits para representar a seqüência de números, tal seqüência cobre uma faixa de $2^m - 1$ possibilidades. Se $m = 3$, por exemplo, a seqüência numérica cobre a faixa de 0 a 7. Entretanto, nada impede a repetição dessa seqüência. Então, a seqüência de número é

$$0, 1, 2, 3, 4, 5, 6, 7, 0, 1, 2, 3, 4, 5, 6, 7, 0, 1, \dots$$

Janela de Transmissão

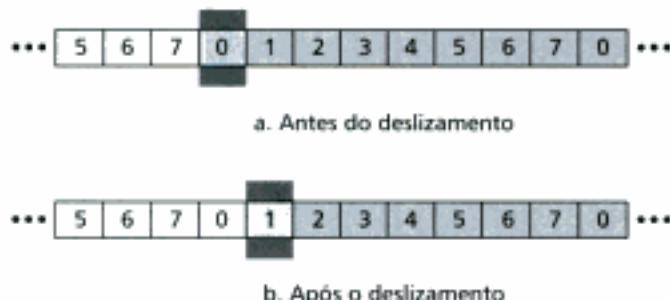
No referencial do transmissor, usamos o conceito de janela para manter ativos em um *buffer* todos os *frames* pendentes até que sejam recebidos os respectivos *frames* de confirmação. Vamos imaginar que todos os *frames* enviados estejam armazenados num *buffer*. Os *frames* aguardam confirmação e permanecem encerrados numa janela. Os *frames* que deixarem a janela são aqueles que realmente foram confirmados, através de um ACK, sendo excluídos automaticamente da fila no *buffer*. Os *frames* à direita da janela não podem ser transmitidos enquanto a janela deslizar sobre eles. Assim, de acordo com a discussão acima, o tamanho da janela é, quando muito, $2^m - 1$.

O tamanho da janela é fixo neste protocolo, embora possamos ter um tamanho de janela variável em outros protocolos, tal como o TCP (veja Capítulo 22). A janela desliza de modo a incluir novos *frames*, que aguardam transmissão, na medida em que ACKs válidos vêm sendo recebidos. Essa janela é denominada **janela móvel (sliding window)**. Por exemplo, na Figura 11.6a, os *frames* de 0 a 6 foram transmitidos. Na figura b, a janela desliza dois *frames* à direita porque os ACKs dos *frames* 0 e 1 foram recebidos.

**Figura 11.6** Janela móvel do transmissor.

Janela de Recepção

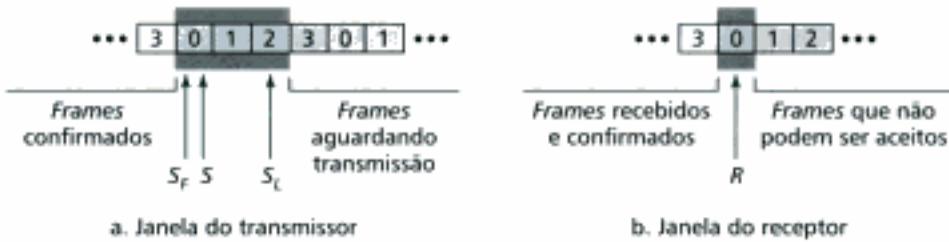
No referencial do receptor, o tamanho da janela é constantemente 1. O receptor está sempre olhando para um *frame* específico, recebido numa ordem específica. Os *frames* recebidos fora de ordem são descartados, sendo colocado o pedido de retransmissão por parte do receptor. A janela de recepção também desliza à direita, conforme ilustra a Figura 11.7. Na figura *a*, o receptor espera pelo *frame* 0. Quando esse *frame* é recebido, a janela é projetada à direita de modo a esperar o *frame* 1.

**Figura 11.7** Janela móvel do receptor.

Variáveis de Controle

Associamos ao transmissor três variáveis de controle, a saber: S , S_F e S_L . A variável S mantém o número de seqüência do *frame* recentemente enviado. A variável S_F mantém o número de seqüência do primeiro *frame* na janela e a variável S_L mantém o número de seqüência do último *frame* na janela. O tamanho da janela é W , onde $W = S_L - S_F + 1$.

O receptor possui uma única variável, denominada por R , que mantém o número de seqüência do *frame* que ele espera receber. Se o número de seqüência do *frame* recebido coincidir com o valor de R , o *frame* é aceito. Senão, o *frame* é rejeitado. A Figura 11.8 mostra as janelas de transmissão e recepção com as respectivas variáveis de controle.

**Figura 11.8** Variáveis de controle.

Relógios

O transmissor dispara um relógio para cada *frame* enviado. O receptor não utiliza relógios.

Confirmação (ACK)

O receptor envia ACK positivo ao transmissor se um *frame* foi recebido com integridade e na devida ordem. Se o *frame* foi corrompido durante a transmissão ou chegou fora de ordem, o receptor mantém-se em silêncio (nenhum ACK), descartando todos os *frames* que chegarem até que o *frame* que ele espera receber seja efetivamente recebido. Esse silêncio do receptor pode fazer com que o relógio do *frame* não-confirmado expire no transmissor, o que força a retransmissão de todos os *frames* que não receberem um ACK na janela. Além disso, o receptor não precisa confirmar cada *frame* isoladamente. Ele pode ser configurado para enviar um único ACK para confirmar todos os *frames* que ele receber num intervalo de tempo.

Frames Retransmitidos

Quando um *frame* é corrompido, o transmissor volta atrás e envia novamente um conjunto de *frames* iniciando no *frame* danificado até o último previsto na seqüência. Por exemplo, suponha numa certa seqüência que o transmissor tenha enviado todos os *frames* até o de número 6, mas o relógio referente ao *frame* 3 expira. Isto significa que o *frame* 3 não teve confirmação recebida, o que força o transmissor a retransmitir novamente os *frames* 3, 4, 5 e 6. Isto acontece porque o protocolo utilizado é o Go-Back-N ARQ.

Operação

Vejamos o que acontece em várias situações.

Operação Normal

A Figura 11.9 ilustra a operação típica do mecanismo Go-Back-N ARQ. O transmissor envia uma fila de *frames* e atualiza as variáveis e janelas à medida que os ACKs vão sendo recebidos.

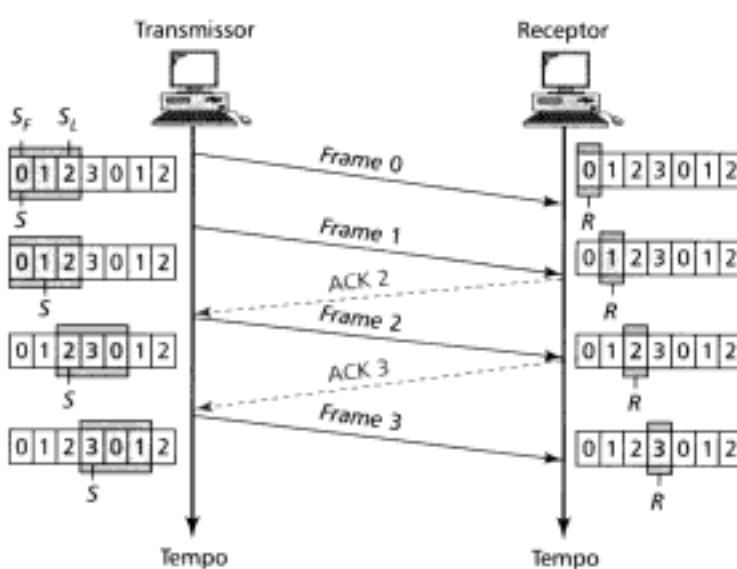


Figura 11.9 Go-Back-N ARQ: operação normal.

Frames Perdidos ou Corrompidos

Vamos verificar o que ocorre se um *frame* é perdido durante a transmissão. A Figura 11.10 ilustra a perda do *frame* de número 2. Quando o receptor receber o *frame* número 3, ele o descarta-

rá porque a variável R aponta para recebimento do frame 2, de acordo com a janela. Expirando o relógio do frame 2, o transmissor retransmite os frames 2 e 3 (ou seja, ele retorna ao frame número 2).

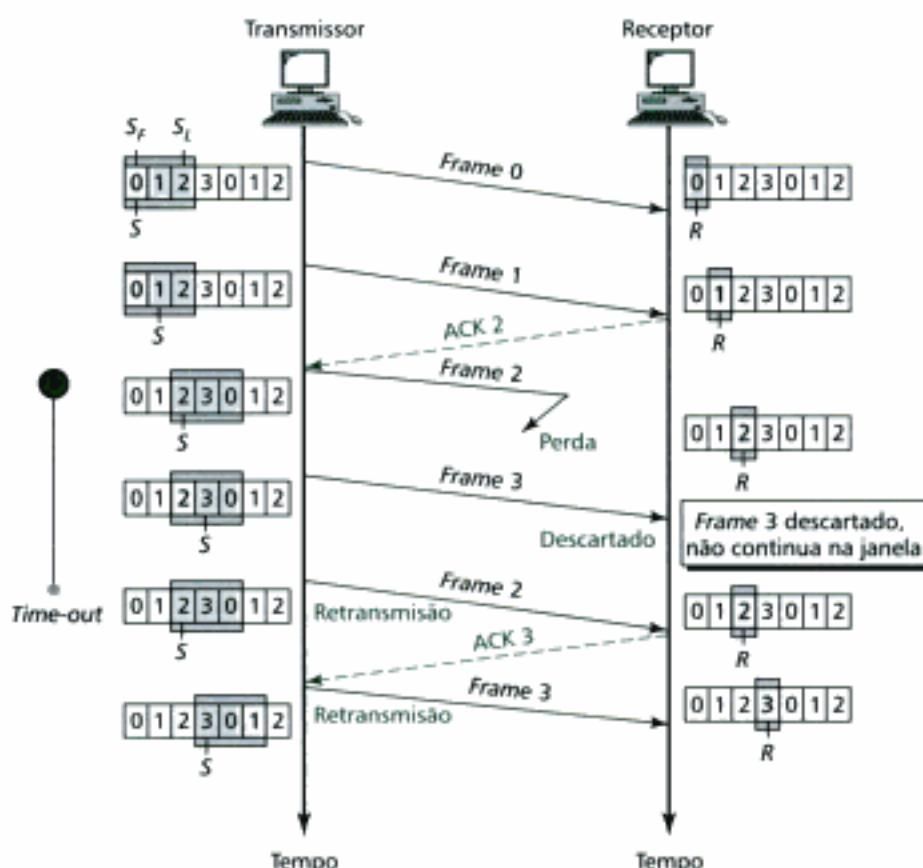


Figura 11.10 Go-Back-N ARQ: perda do frame.

ACK Perdido ou Corrompido

Podem acontecer duas situações se uma resposta ACK for perdida ou corrompida. Se o ACK correspondente à próxima confirmação for recebido antes da expiração de qualquer relógio do transmissor, não será necessário retransmitir os frames porque os ACKs são cumulativos neste protocolo. Uma resposta ACK 4 significa responder os frames de 1 a 4. Então, se os ACKs 1, 2 e 3 forem perdidos, o ACK 4 pode confirmá-los. Entretanto, se o próximo ACK chegar após o *time-out* do relógio transmissor, o frame específico e todos os frames que o sucederem serão reenviados. Note que um receptor nunca retransmite ACKs. Deixaremos a figura e os detalhes como um exercício ao estudante.

Atraso da Confirmação (ACK)

O atraso da confirmação (ACK) também força a retransmissão dos frames. Outra vez, deixaremos ao estudante os detalhes e a tarefa de analisar a figura.

Tamanho da Janela de Transmissão

Estamos prontos para justificar o fato do tamanho da janela de transmissão ser no mínimo 2^n . Por exemplo, vamos escolher $m = 2$. Isto significa que o tamanho da janela deve ser $2^n - 1 = 3$. A Figura 11.11 compara janelas de tamanho 3 e 4, respectivamente.

Se o tamanho da janela é 3 (menor que 2^2) e os três ACKs são perdidos, o relógio do frame 0 expira e todos os frames são retransmitidos. Contudo, a janela de receção está posicionada de mo-

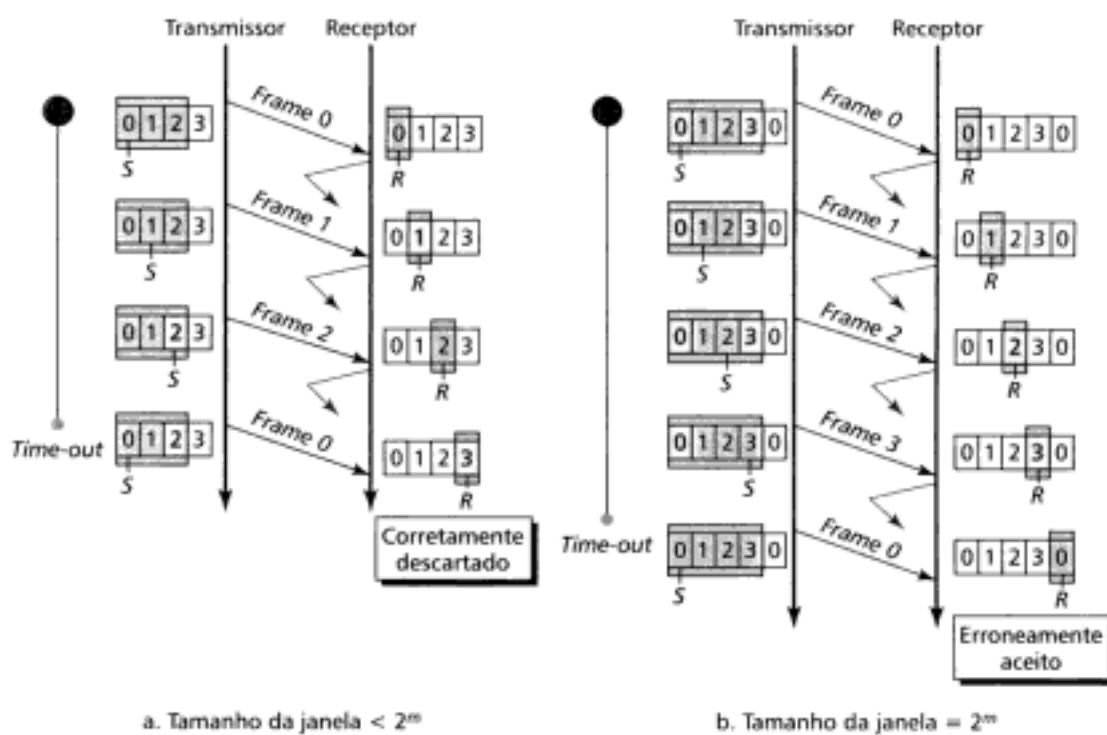


Figura 11.11 Go-Back-N ARQ: tamanho da janela de transmissão.

do a receber o *frame* 3, não o *frame* 0. Assim, os *frames* duplicados são corretamente descartados. De outro modo, se o tamanho da janela é 4 (igual a 2^2) e todos os ACKs são perdidos, o transmissor reenvia uma duplicata do *frame* 0. Desta vez, a janela de recepção está posicionada a receber o *frame* 0, então ele o recebe, não como uma duplicata, mas como o primeiro *frame* do próximo ciclo. Isto constitui um erro.

No protocolo Go-Back-N ARQ, o tamanho da janela de transmissão deve ser menor que 2^m . A janela de recepção tem sempre tamanho unitário.

Transmissão Bidirecional e Piggybacking

Assim como no caso do protocolo Stop-and-Wait ARQ, o protocolo Go-Back-N também pode ser bidirecional. Logo, podemos utilizar o conceito de *piggybacking* para melhorar a eficiência da transmissão. Todavia, note que para cada sentido de transmissão é necessário o conceito de janela tanto no transmissor quanto no receptor. Deixaremos a análise dessa configuração ao estudante, como um exercício.

11.4 SELECTIVE REPEAT ARQ

O mecanismo Go-Back-N ARQ simplifica o processo de comunicação do ponto de vista do receptor. O receptor mapeia a recepção através de uma única variável, dispensando um *buffer* para manter os *frames* desordenados, que são simplesmente descartados. Contudo, este protocolo é muito inefficiente num *link* ruidoso. A probabilidade de um *frame* ser danificado ou corrompido num *link* ruidoso é muito alta. Isso implica numa elevada taxa de retransmissão de *frames* múltiplos. Reenviar todos os *frames* requer muita banda e diminui a velocidade da transmissão. Para tratar desses casos, existe outro mecanismo que não reenvia *N frames* quando apenas um está danificado. Somente o *frame* danificado é retransmitido. Este mecanismo é denominado Selective Repeat ARQ. Ele é muito eficiente em *links* ruidosos, mas torna mais complexo o processamento dos *frames* no receptor.

Janelas de Transmissão e Recepção

A configuração do transmissor e as respectivas variáveis de controle para o mecanismo Selective Repeat ARQ são os mesmos discutidos para o Go-Back-N ARQ. Entretanto, conforme discutiremos adiante, o tamanho da janela de transmissão deve ser, quando muito, a metade do valor 2^m . A janela de recepção também deve ter este tamanho. Porém, essa janela especifica a faixa de frames recebidos com sucesso. Noutras palavras, no mecanismo Go-Back-N o receptor olha para um número de sequência específico e, no protocolo Selective Repeat, o receptor olha para uma faixa de números de sequência. O receptor é dotado de duas variáveis de controle, denotadas por R_f e R_L , que define as fronteiras ou tamanho da janela. A Figura 11.12 ilustra as janelas no lado do transmissor e do receptor.

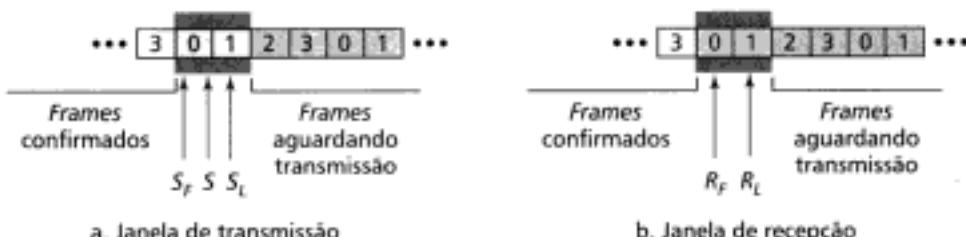


Figura 11.12 Selective Repeat ARQ: janela de transmissão e recepção.

Para o mecanismo Selective Repeat também está definida a confirmação sem resposta **Negative Acknowledgment (NAK)** que reporta, antes do tempo de expiração do relógio, o número de sequência de um frame danificado.

Operação

Exemplificaremos a operação desse mecanismo através da perda de frame, conforme ilustra a Figura 11.13.

Os frames 0 e 1 são aceitos no ato do recebimento porque estão dentro da faixa especificada pela janela de recepção. Quando o frame 3 for recebido, ele é aceito pelo mesmo motivo. Entretanto, o receptor envia um NAK 2 para mostrar que o frame 2 não foi recebido. Ao receber o NAK 2, o transmissor reenvia somente o frame 2, o que pode prontamente ser aceito pelo receptor porque está na faixa coberta pela janela desse dispositivo.

ACKs e NAKs Perdidos ou Atrasados

Deixaremos a perda e atrasos de ACKs e NAKs como exercícios. Para analisá-los, perceba que o transmissor também dispara um relógio para cada frame enviado.

Tamanho da Janela de Transmissão

Estamos prontos a mostrar porque o tamanho das janelas de transmissão e recepção devem ser, quando muito, metade de 2^m . Vamos fazer $m = 2$, como exemplo, o que significa que o tamanho da janela deveria ser $2^m/2 = 2$. A Figura 11.14 compara uma janela de tamanho 2 e 3, respectivamente.

Se o tamanho da janela vale 2 e todos os ACKs são perdidos, o relógio para o frame 0 expira e ocorre a retransmissão desse frame. Contudo, a janela de recepção espera pelo frame 2, não o frame 0. Então, o receptor interpreta o frame 0 como uma duplicata e o descarta corretamente. Quando o tamanho da janela vale 3 e todos os ACKs são perdidos, o transmissor reenvia uma duplicata do frame 0. Porém, dessa vez, a janela de recepção aponta para o frame 0 (0 é parte da janela). Então, o receptor aceita o frame 0, não como uma duplicata, mas como o primeiro frame do próximo ciclo. Isto constitui claramente um erro.

No protocolo Selective Repeat ARQ, o tamanho das janelas de transmissão e de recepção deve ser, quando muito, $2^m/2$.

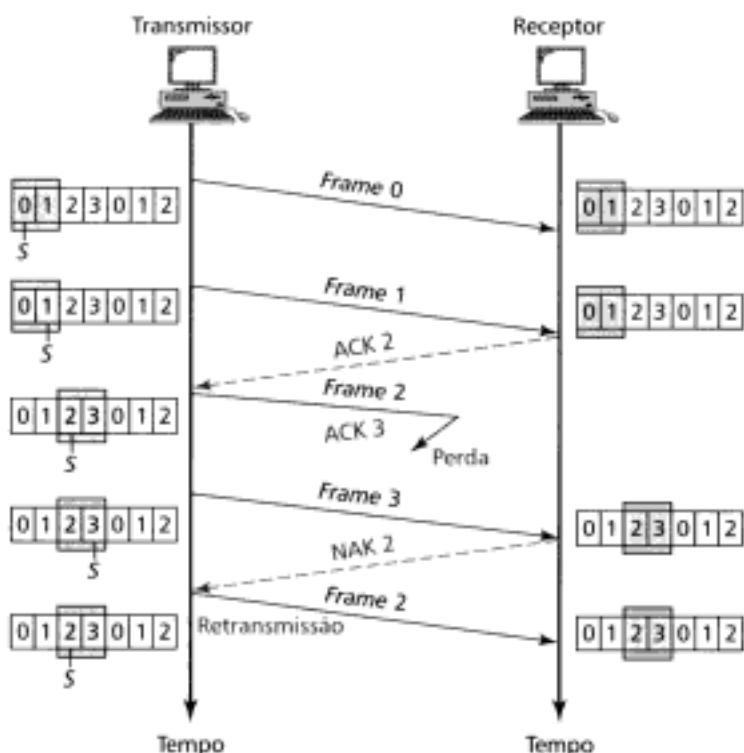


Figura 11.13 Selective Repeat ARQ: perda do frame.

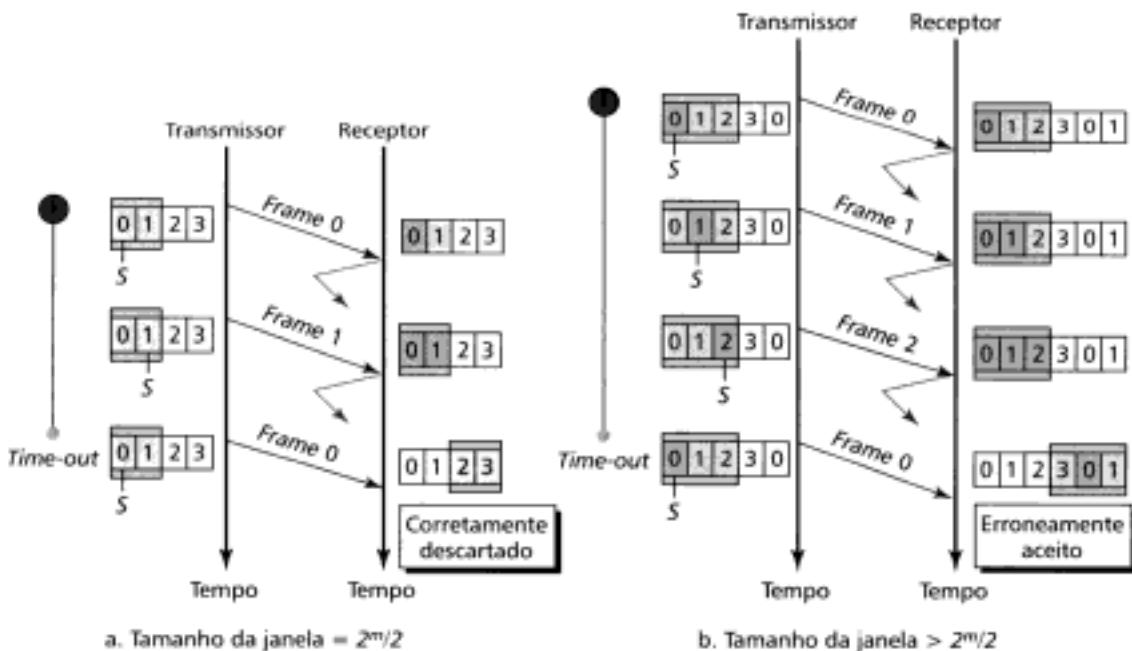


Figura 11.14 Selective Repeat ARQ: tamanho da janela de transmissão.

Transmissão Bidirecional e Piggybacking

Como nos casos do Stop-and-Wait ARQ e Go-Back-N ARQ, o mecanismo Selective Repeat ARQ também pode lidar com transmissões bidirecionais. Podemos utilizar a técnica de *piggybacking* para melhorar a eficiência da transmissão. Todavia, note que para cada sentido de transmissão é necessário o conceito de janela tanto no transmissor quanto no receptor. Deixaremos a análise dessa configuração ao estudante, como um exercício.

Produto Banda × Tempo de Propagação

Uma medida da eficiência do sistema ARQ é o produto da largura de banda (em *bits* por segundo) pelo tempo de propagação de ida e volta dos *frames* (em segundos). Se o *link* tiver uma largura de banda adequada, a janela de transmissão poderá ser exaurida rapidamente o transmissor terá que esperar os ACKs de retorno. Se o tempo de propagação é grande, o tempo de espera nos relógios dos *frames* do transmissor podem exaurir a janela enquanto espera pelos ACKs. Então, o produto desses fatores dá uma medida da eficiência do sistema ARQ. O **produto banda × tempo de propagação** é uma medida do número de *bits* que um sistema transmissor pode enviar enquanto aguarda notícias do receptor.

Exemplo 1

No sistema Stop-and-Wait ARQ, a largura de banda de uma linha vale 1 Mbps e 1 *bit* leva 20 ms de viagem de ida e volta. Qual o produto banda × tempo de propagação? Se os *frames* de dados têm 1000 *bits* de tamanho, qual é o percentual de utilização do *link*?

Solução

O produto banda × tempo de propagação é

$$1 \times 10^6 \times 20 \times 10^{-3} = 20.000 \text{ bits}$$

Este sistema pode transmitir 20.000 *bits* durante o intervalo de tempo de ida e volta dos *frames* de dados do transmissor ao receptor. Entretanto, o sistema transmite somente 1000 *bits* por vez. Assim, podemos dizer que o percentual de utilização do *link* é de 1.000/20.000 ou 5%. Por este motivo, o protocolo Stop-and-Wait é uma ótima escolha quando se deseja evitar o desperdício de banda em *links* banda larga ou quando estiverem envolvidos tempos de propagação elevados.

Exemplo 2

No Exemplo 1, qual é o percentual de utilização do *link* se o método adotado for Go-Back-N com uma sequência de 15 *frames*?

Solução

O produto banda × tempo de propagação ainda vale 20.000. O sistema consegue transmitir 15 *frames* ou 15.000 *bits* durante o intervalo de tempo de ida e volta dos *frames* de dados do transmissor ao receptor. Isto representa um percentual de 15.000/20.000 ou 75%. Claro, se ocorrem danos aos *frames* durante o processo de transmissão de dados, o percentual de utilização será inferior ao valor supra calculado porque haverá retransmissão de *frames*.

Pipelining

Tanto nas redes de computadores quanto em outras áreas, uma tarefa é freqüentemente iniciada antes que a anterior tenha sido finalizada. Isto é conhecido como **pipelining**. Não há *pipelining* no sistema Stop-and-Wait ARQ porque o transmissor precisa esperar até que o *frame* alcance o destino e seja reconhecido antes do próximo ser colocado no *link*. A técnica de *pipelining* aplica-se aos protocolos Go-Back-N ARQ e Selective Repeat ARQ porque muitos *frames* podem ser enviados antes do receptor dar alguma resposta sobre os *frames* anteriores.

A técnica de *pipelining* melhora a eficiência da transmissão, se o número de *bits* na transição for grande em relação ao produto banda × tempo de propagação.

11.5 HDLC

Protocolo padrão ISO, o **High-level Data Link Control (HDLC)** é um protocolo desenvolvido para suportar comunicação tanto em modo *half-duplex* quanto em *full-duplex*, através de *links* ponto a ponto ou multiponto. O protocolo HDLC implementa os mecanismos ARQ discutidos neste capítulo.

Configurações e Modos de Transferência

O HDLC proporciona transmissão em dois modos, a saber: NRM e ABM.

NRM

No modo **NRM (Normal Response Mode)**, a configuração da estação é desbalanceada ou assimétrica. Nesse modo, temos duas estações, uma principal e outra secundária. Uma **estação principal** pode apenas enviar comandos enquanto a **estação secundária** só pode responder aos comandos. O modo NRM é utilizado em *links* ponto a ponto ou multiponto. Veja a Figura 11.15.

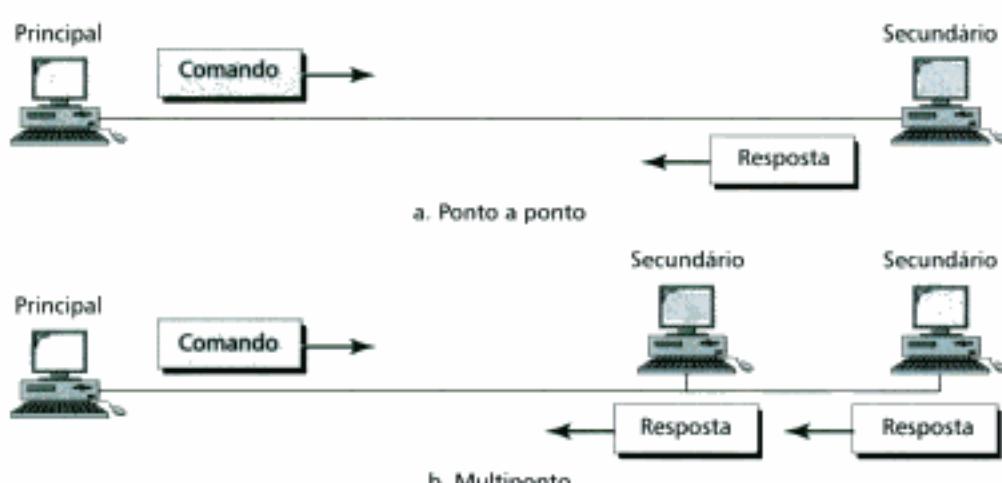


Figura 11.15 NRM.

ABM

No modo **ABM (Asynchronous Balanced Mode)**, a configuração da estação é balanceada. O *link* é ponto a ponto e cada estação é tanto principal quanto secundária, conforme ilustra a Figura 11.16.



Figura 11.16 ABM.

Frames

O protocolo HDLC define três tipos de *frames* para prover a flexibilidade necessária e suportar todas as possíveis opções nos modos e nas configurações descritas acima: **I-frames (frames de informação)**, **S-frames (frames de supervisão)** e **U-frames (frames não numerados)**. Cada tipo de *frame* funciona como um envelope para transmissão de diferentes tipos de mensagens. Os I-frames são utilizados para transportar dados do usuário e informação de controle relativa aos dados (*piggybacking*). Os S-frames são utilizados com um único propósito: transportar informação de controle. Os U-frames são reservados ao gerenciamento do sistema. A informação transportada pelos U-frames serve para gerenciar o uso do *link*.

Formato do Frame

Cada *frame* do protocolo HDLC pode conter seis campos, conforme Figura 11.17. O *frame* é constituído basicamente de: campos *flag* (bandeira) de início e fim, campo de endereço, campo de con-

trole, campo de informação e campo FCS. Nas transmissões de múltiplos *frames*, o campo *flag* de fim de um *frame* pode servir como *flag* de início do próximo.

Campo Flag

O **campo flag** do protocolo HDLC é uma sequência de 8-bits, cujo padrão é 01111110, que identifica tanto o início quanto o final de um *frame* e serve para sincronizar receptor e transmissor. O campo *flag* é discutido mais detalhadamente na seção sobre transparência da transmissão de dados.

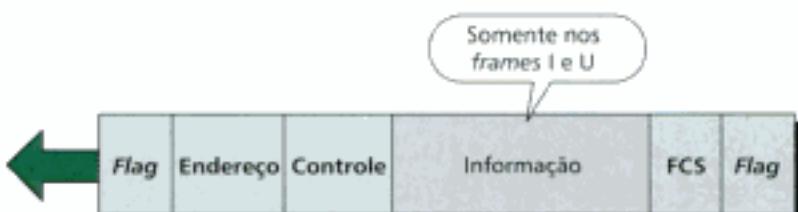


Figura 11.17 Frame HDLC.

Campo de Endereço

O segundo campo de um *frame* HDLC endereça a estação secundária que é a origem ou destino do *frame*, ou talvez, uma estação funcionando como estação secundária no caso de combinação das estações. Se a estação principal montar o *frame*, o campo de endereço desse *frame* contém um endereço de destino. Caso contrário, o campo de endereço do *frame* contém um endereço de origem. O tamanho de um **campo de endereço** pode ser um ou muitos bytes, dependendo do tipo e tamanho da rede. Um byte identifica univocamente 128 estações, porque um bit é utilizado para outro propósito. Redes muito grandes requerem muitos bytes de identificação no campo de endereços.

Se o campo de endereço possuir apenas um byte, o último bit será sempre 1. Se o endereço tiver mais de um byte, todos os bytes terminarão com 0, exceto o último, que terminará com 1. O final zero (0) em cada byte intermediário indica ao receptor que ainda existem mais bytes de endereço para chegar. Redes que não utilizam a configuração principal/secundária, tais como a Ethernet (Capítulo 14), usam dois campos de endereço: um campo de endereço origem e outro de endereço de destino.

Campo de Controle

O **campo de controle** é um segmento do *frame* de 1 ou 2 bytes utilizados para controle de fluxo e erros. A interpretação dos bits deste campo é diferente para diferentes tipos de *frames*. Discutiremos este campo quando tratarmos os tipos de *frames*.

Campo de Informação

O **campo de informação** contém os dados na camada ou nível de rede ou informação sobre o gerenciamento da rede. O tamanho desse campo pode variar, dependendo da rede, mas é sempre fixo dentro de uma mesma rede.

Campo FCS

O campo de detecção de erro do protocolo HDLC é o **FCS (frame check sequence)**. Este campo pode conter 2 ou 4 bytes de CRC (padrão ITU-T).

Tipo de Frame

O protocolo HDLC também define três tipos de *frames*: I-frames, S-frames e U-frames, conforme ilustra a Figura 11.18.

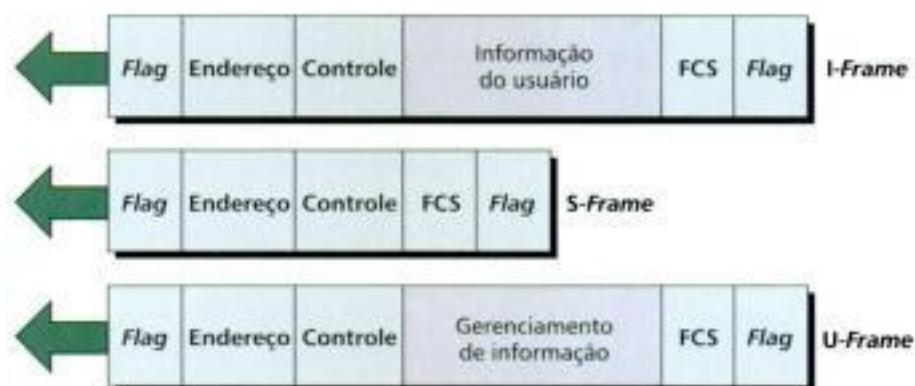


Figura 11.18 Tipos de frames HDLC.

I-Frame

Os I-frames foram desenvolvidos para transportar dados da camada de rede das estações. Além disso, os I-frames podem incluir mecanismos de controle de fluxo e erro (*piggybacking*). A Figura 11.19 mostra a estrutura do campo de controle de um I-frame.

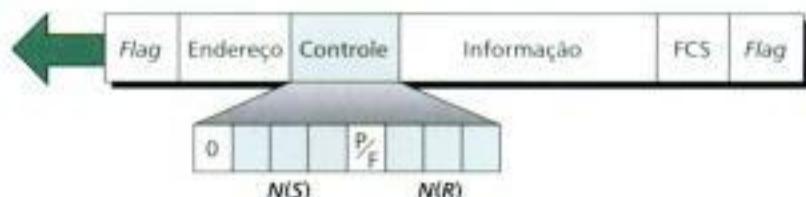


Figura 11.19 I-frame.

No campo de controle do I-frame, os bits devem ser interpretados da seguinte forma:

- Se o primeiro bit do campo de controle vale 0 é sinalizado um I-frame na rede.
- Os próximos 3-bits, denominados $N(S)$, definem o número de seqüência do frame que estiver em trânsito. Três bits podem definir uma seqüência de números entre 0 e 7. O valor deste campo corresponde ao valor da variável de controle S , discutida nos três mecanismos de ARQ.
- O próximo bit é denominado bit P/F. O campo P/F possui dois propósitos. Ele tem significado somente quando está ativo (bit = 1) e pode significar uma consulta (*poll*) ou final. Ele significa uma consulta P (*poll*) quando o frame é enviado da estação principal para a secundária, isto é, quando o campo de endereço contém o endereço do receptor. O estado final F ocorre quando o frame é enviado da estação secundária para a principal (quando o campo de endereço traz o endereço do transmissor).
- Os próximos 3-bits, denominados $N(R)$, correspondem ao valor do ACK quando a técnica de *piggybacking* é utilizada.

S-Frames

Os frames de supervisão são utilizados para controle de fluxo e erro sempre que é impossível ou inapropriado utilizar o *piggybacking* (quando a estação não tem dados próprios a transmitir ou necessita enviar um comando/resposta diferente de um ACK). Os S-frames não possuem campo de informação. A Figura 11.20 ilustra o formato do campo de controle para um S-frame.

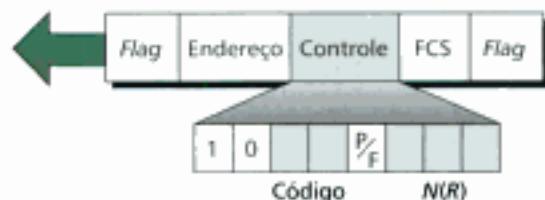


Figura 11.20 Campo controle no S-frame HDLC.

No campo de controle do S-frame, os *bits* devem ser interpretados da seguinte forma:

- Se os primeiros 2-*bits* do campo de controle estiverem em 10 é sinalizado um S-frame na rede.
- Os próximos 2-*bits* do campo de controle de um S-frame codificam quatro tipos de S-frames, a saber: receptor pronto (*Receive Ready* – RR), receptor ocupado (*Receive Not Ready* – RNR), rejeitado (*Reject* – REJ) e rejeição seletiva (*Selective Reject* – SREJ).
 - a. **RR**: se o valor desses 2-*bits* de código é 00, o S-frame sinaliza receptor pronto. É usado pelo receptor para confirmar o recebimento de I-frames quando este não tem I-frames para transmitir, estando a estação pronta para receber novos frames. Os frames até $N(R) - 1$ foram recebidos corretamente.
 - b. **RNR**: se o valor desses 2-*bits* de código é 10, o S-frame sinaliza receptor ocupado. É usado para controlar o fluxo de informação, ordenando ao transmissor que interrompa momentaneamente a transmissão pois o receptor não possui mais espaço em buffer para enfileirar e armazenar frames. Este é essencialmente um frame de controle de fluxo.
 - c. **REJ**: se o valor desses 2-*bits* de código é 01, o S-frame sinaliza rejeição ou perda de frames (possivelmente perda do número de seqüência). Este é um NAK frame, mas não é semelhante aquele estudado no protocolo Selective Repeat ARQ. Este é um NAK frame utilizado no protocolo Go-Back-N para melhoria do processo de informação do transmissor, comunicando sobre uma perda ou dano do último frame, antes que o relógio do transmissor expire. Os frames até $N(R) - 1$ foram recebidos corretamente.
 - d. **SREJ**: se o valor desses 2-*bits* de código é 11, o S-frame sinaliza rejeição seletiva. Este é um NAK frame usado no protocolo Selective Repeat ARQ. Note que o protocolo HDLC utiliza o termo Selective Reject no lugar de Selective Repeat.
- O quinto bit no campo de controle é o bit P/F discutido anteriormente.
- Os próximos 3-*bits*, denominados $N(R)$, correspondem aos valores ACK ou NAK, conforme o caso.

U-Frames

Os frames não numerados (U-frames) são utilizados durante as permutas das sessões de frames de gerenciamento e controle de informação entre dispositivos. Diferentemente dos S-frames, os U-frames possuem um campo de informação, mas o utilizam para gerenciamento do sistema de informações, não para troca de dados do usuário. Entretanto, assim como os S-frames, a maior parte da informação transportada pelos U-frames aparece codificada no campo de controle. Os códigos dos U-frames são divididos em duas seções: um prefixo de 2-*bits* precedendo o bit P/F e um sufixo de 3-*bits* sucedendo o bit P/F. Juntas, estas seções perfazem 5-*bits*, gerando 32 tipos diferentes de U-frames. Algumas das combinações mais comuns são mostradas na Figura 11.21.

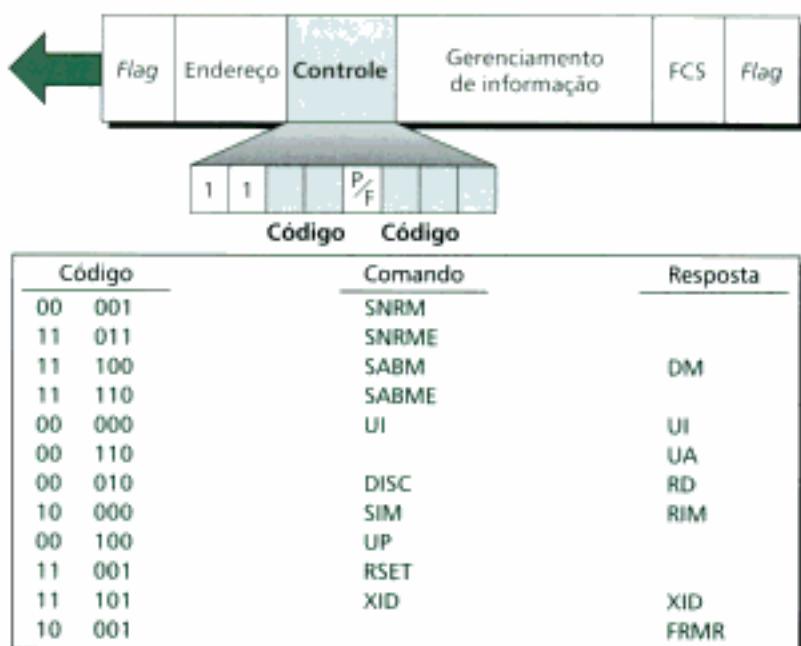


Figura 11.21 Campo de controle no U-frame HDLC.

Os diferentes tipos de U-frames e os respectivos significados estão listados na Tabela 11.1. Perceba que eles podem ser utilizados para muitos propósitos diferentes, tais como seleção de modo, trocas de frames não numerados, conexão e desconexão do link (entre outros).

TABELA 11.1 Comando e resposta U-Frame

Comando/resposta	Significado
SNRM	normal response mode
SNRME	Set normal response mode (extended)
SABM	Set asynchronous balanced mode
SABME	Set asynchronous balanced mode (extended)
UP	Unnumbered poll
UI	Unnumbered information
UA	Unnumbered acknowledgment
RD	Request disconnect
DISC	Disconnect
DM	Disconnect mode
RIM	Request information mode
SIM	Set initialization mode
RSET	Reset
XID	Exchange ID
FRMR	Frame reject

Exemplos

Nesta seção, mostraremos alguns exemplos de processos de comunicação usando o protocolo HDLC.

Exemplo 3: piggybacking sem erro

A Figura 11.22 mostra uma negociação e troca de frames entre duas estações.

A estação A inicia a troca de informação utilizando um I-frame número 0, seguido de outro I-frame número 1. A estação B superpõe (piggybacks) os ACKs desses dois frames em um único I-frame, transmitindo-os

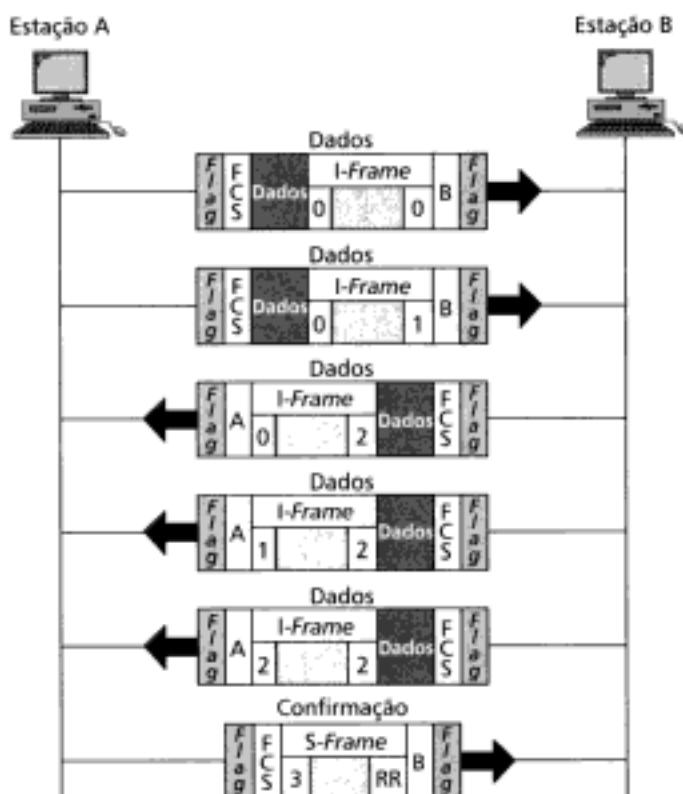


Figura 11.22 Exemplo 3.

à estação A. O primeiro *I-frame* da estação B é identificado com o número 0 [campo $N(S)$] e contém um 2 no campo $N(R)$, confirmando (ACK) o recebimento dos *frames* 0 e 1 da estação A, além de indicar que espera receber o *frame* número 2. A estação B transmite um segundo e terceiros *I-frames*, cujos números são 1 e 2, antes de voltar a aceitar *frames* da estação A. Assim, a informação do campo $N(R)$ da estação B não foi modificada: os *frames* 1 e 2 da estação B indicam que ela ainda espera receber o *frame* número 2 da estação A.

A estação A tinha enviado todos os dados dela. Então, ela não pode superpor (*piggyback*) um ACK num *I-frame* e enviar como um *S-frame*. O código RR indica que A ainda está pronta para receber. O número 3 no campo $N(R)$ informa a B que os *frames* 0, 1 e 2 foram aceitos e que A aguarda o *frame* número 3.

Exemplo 4: piggybacking com erro

No Exemplo 3, suponha que o *frame* 1, enviado da estação B para a estação A, chegue com erro. A estação A informa à estação B para retransmitir os *frames* 1 e 2 (o sistema usa o mecanismo *Go-Back-N*). Dessa forma, a estação A envia um *S-frame* de rejeição para informar o erro no *frame* 1. A Figura 11.23 ilustra esta troca de *frames*.

Transparência de Dados

O campo de dados do *frame* HDLC pode transportar texto assim como informação não textual, tal como gráficos, áudio, vídeo e outras seqüências de bits. Infelizmente, alguns tipos de mensagens podem criar alguns problemas durante a transmissão. Por exemplo, se o campo de dados de um *frame* HDLC contém um padrão idêntico à seqüência reservada para o campo *flag* (01111110), o receptor interpreta essa seqüência como *flag* de fim de transmissão. O restante dos bits são assumidos como parte do próximo *frame*. Este fenômeno é denominado falta ou perda de **transparência de dados**. Quando os dados são transparentes, toda massa de dados é reconhecida efetivamente como dados e, por extensão, toda informação de controle é reconhecida como informação de controle.

Bit de enchimento (stuffing)

Para assegurar que a seqüência do campo *flag* não se repita inadvertidamente em qualquer outro lugar do *frame*, o protocolo HDLC utiliza **bit de enchimento (stuffing)**. Toda vez que o transmissor tiver uma seqüência de dados com mais de 5-bits (nível 1) consecutivos, é inserido

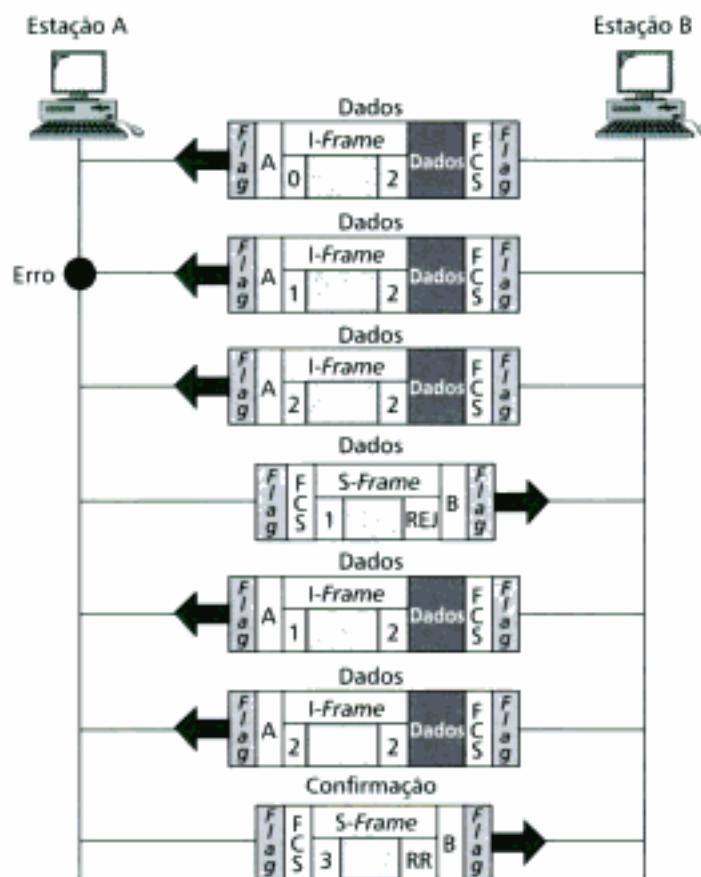


Figura 11.23 Exemplo 4.

um zero (0) redundante após o quinto bit 1. Por exemplo, a seqüência 01111111000 torna-se 0111110111000. O bit 0 extra é inserido independentemente do sexto bit ser outro bit 1. A presença desse 0 informa ao receptor que essa seqüência não é um flag de fim. Uma vez recebido no receptor, o zero extra (o enchimento) é retirado dos dados e o bit original é restabelecido.

Stuffing (bit de enchimento) é nome do processo de adição de um 0 extra à informação sempre que existir mais de 5-bits 1s consecutivos na seqüência de dados, de modo que o receptor não cometa erro, interpretando dados como informação de flag.

A Figura 11.24 ilustra o bit de enchimento sendo inserido no transmissor e removido no receptor. Note que, até mesmo se tivermos um 0 após o 5-bit 1, o enchimento ainda será feito com 0. Este zero extra foi removido no receptor.

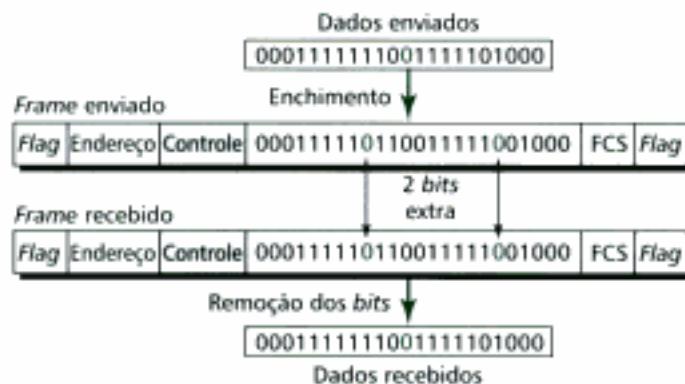


Figura 11.24 Bit de enchimento (stuffing).

O bit de enchimento é necessário toda vez que ocorrerem 5-bits 1s consecutivos. As três exceções são quando a seqüência de bits for verdadeiramente um flag, quando a transmissão estiver sendo abortada e quando o canal estiver ocioso. O fluxograma na Figura 11.25 ilustra o processo utilizado pelo receptor para identificar e descartar um bit de enchimento. O fluxo baseia-se na leitura dos bits de chegada no receptor. Ele basicamente conta os 1s. Sempre que encontrar 5-bits 1s consecutivos após um 0, verifica o próximo bit (o sétimo). Se o sétimo bit é um 0, o receptor reconhece que foi introduzido um bit de enchimento, o descarta e reseta o contador. Se o sétimo bit é um 1, o receptor verifica o oitavo bit. Se o oitavo é um 0, a seqüência é reconhecida como um flag e é tratada como tal. Se o oitavo bit é um 1, o receptor continua o processo de contagem. Um total de 7 a 14-bits 1s consecutivos indicam uma operação abortada. Um total de 15 ou mais bits indicam canal ocioso.

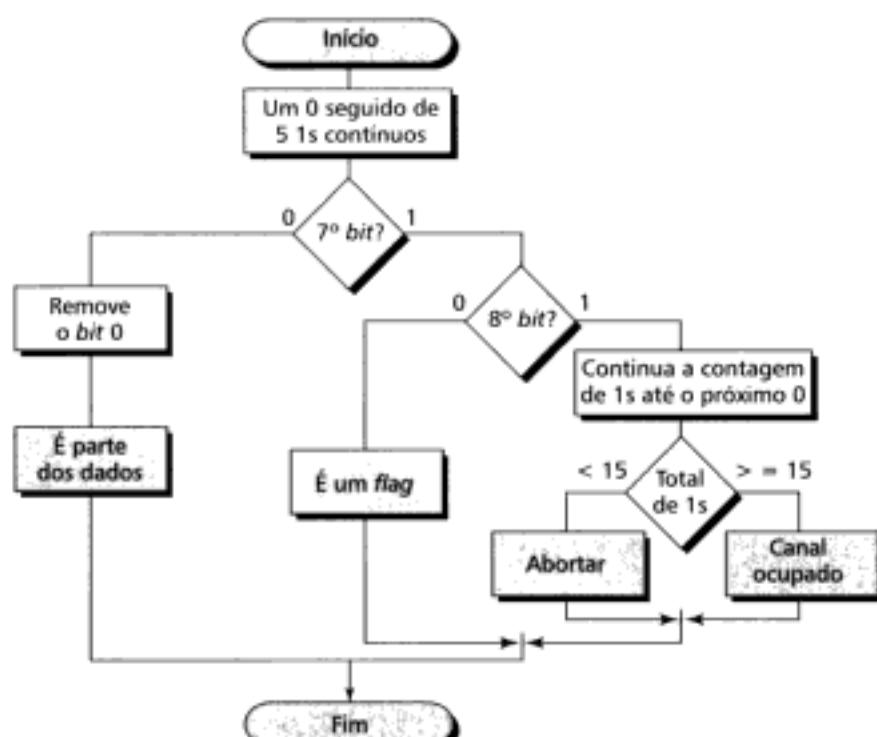


Figura 11.25 Stuffing no HDLC.

11.6 TERMOS-CHAVE

Asynchronous Balanced Mode (ABM)	Frame check sequence (FCS)
Automatic Repeat Request (ARQ)	Frame de informação (<i>I-frame</i>)
Bit de enchimento (stuffing)	Frame de supervisão (<i>S-frame</i>)
Campo de endereço	Frame não numerado (<i>U-frame</i>)
Campo de informação	Go-Back-N ARQ
Campo flag	High-level Data Link Control (HDLC)
Confirmação (ACK)	Janela móvel
Confirmação sem resposta (Negative Acknowledgment – NAK)	Normal Response Mode (NRM)
Controle de erro	Piggybacking (superposição)
Controle de fluxo	Pipelining
Controle do link ou enlace de dados	Produto banda × tempo de propagação
Estação principal	Selective Repeat ARQ
Estação secundária	Stop-and-Wait ARQ
	Transparência de dados

11.7 RESUMO

- Controle de fluxo ordena a taxa de transmissão do transmissor para que ele não estoure o *buffer* do receptor, inundando-o de dados.
- Controle de erro é simultaneamente um método de detecção e correção de erros.
- No protocolo Stop-and-Wait ARQ, o transmissor envia um *frame* e espera pelo ACK do receptor antes de enviar o próximo *frame* da sequência.
- No protocolo Go-back-N ARQ, muitos *frames* podem ser enviados e trafegar no meio de transmissão ao mesmo tempo. Se houver um erro, a retransmissão começará do último *frame* não confirmado, até mesmo se os *frames* subsequentes chegaram corretamente. *Frames* duplicados são descartados.
- No protocolo Selective Repeat ARQ, muitos *frames* podem ser enviados e trafegar no meio de transmissão ao mesmo tempo. Se acontecer um erro, o único *frame* retransmitido será o *frame* perdido ou danificado.
- Os mecanismos de controle de fluxo que utilizam o conceito de janela móvel têm variáveis de controle tanto no transmissor quanto no receptor.
- A técnica de *piggybacking* acopla ou superpõe um ACK a um *frame* de dados.
- O produto banda × tempo de propagação dá uma medida da quantidade de *bits* que um sistema pode transportar.
- O protocolo HDLC implementa mecanismos ARQ. Ele suporta comunicação tanto em *links* ponto a ponto quanto em *links* multiponto.
- Estações HDLC se comunicam nos modos NRM (Normal Response Mode) ou ABM (Asynchronous Balanced Mode).
- O protocolo HDLC define três tipos de *frames*: o *frame* de informação (*I-frame*), o *frame* de supervisão (*S-frame*) e o *frame* não numerado (*U-frame*).
- O protocolo HDLC controla a transparência de dados adicionando um 0 sempre que 5-*bits* 1s consecutivos seguem um zero. Este processo de adição de 0 é denominado *bit* de enchimento.

11.8 PRATIQUE OS CONHECIMENTOS ADQUIRIDOS

Questões de Revisão

1. Por que uma comunicação necessita de controle de fluxo?
2. Quais são os três mecanismos típicos de ARQ?
3. Como um ARQ corrige um erro?
4. O mecanismo Stop-and-Wait ARQ possui duas variáveis de controle: *S* e *R*. Quais as funções dessas variáveis?
5. Como o Go-Back-N ARQ difere do Selective Repeat ARQ?
6. Qual é o propósito do relógio transmissor num sistema usando ARQ?
7. Compare os tamanhos das janelas móveis do transmissor e do receptor no protocolo Go-Back-N ARQ.
8. Como acontecem as perdas de ACK e *frame* no transmissor?
9. Compare os tamanhos das janelas móveis do transmissor e do receptor no protocolo Selective Repeat ARQ.
10. Que mecanismo (protocolo) ARQ utiliza *pipelining*?
11. Como o produto banda × tempo de propagação está relacionado à eficiência do sistema?
12. No protocolo HDLC, o que é *bit* de enchimento? Por que ele é necessário?
13. Cite os nomes dos tipos de *frames* HDLC e descreva-os sucintamente.
14. Cite os *bits* do campo de controle HDLC e descreva-os sucintamente.
15. O que significa *piggybacking*?
16. Cite os quatro tipos de *S-frames*.

Questões de Múltipla Escolha

17. No protocolo Go-Back-N ARQ, se o tamanho da janela é 63, qual é a faixa de números de seqüência?
- 0 a 63
 - 0 a 64
 - 1 a 63
 - 1 a 64
18. Controle de fluxo é necessário para prevenir _____.
- Erros de bit
 - Estouro do *buffer* transmissor
 - Estouro do *buffer* receptor
 - Colisão entre o transmissor e o receptor
19. No protocolo Go-Back-N ARQ, se os frames 4, 5 e 6 são recebidos com sucesso, o receptor pode enviar um ACK _____ para o transmissor.
- 5
 - 6
 - 7
 - Todas acima
20. Para uma janela móvel de tamanho $n - 1$ (n número seqüencial), pode haver um máximo de _____ frames enviados e não confirmados.
- 0
 - $n - 1$
 - n
 - $n + 1$
21. No protocolo _____ ARQ, se um NAK é recebido, só o frame perdido ou danificado é retransmitido.
- Stop-and-Wait
 - Go-Back-N
 - Selective Repeat
 - (a) e (b)
22. ARQ significa _____.
- Automatic Repeat Quantization
 - Automatic Repeat Request
 - Automatic Retransmission Request
 - Acknowledge Repeat Request
23. Um relógio é disparado quando _____ é (são) enviado(s).
- Um *frame* de dados
 - Um ACK
 - Um NAK
 - Todas acima
24. No protocolo Stop-and-Wait ARQ, para n pacotes de dados enviados, são necessários _____ ACKs.
- n
 - $2n$
 - $n - 1$
 - $n + 1$
25. HDLC é um acrônimo para _____.
- High-Duplex Line Communication
 - High-Level Data Link Control
 - High-Duplex Digital Link Combination
 - Host Double-Level Circuit
26. O campo de endereço de um *frame* no protocolo HDLC contém o endereço da estação _____.
- Principal
 - Secundária
 - Terciária
 - (a) ou (b)
27. O campo _____ do protocolo HDLC marca o início e o fim de um *frame*.
- Flag
 - Endereço
 - Controle
 - FCS
28. O que está presente em todos os campos de controle no protocolo HDLC?
- Bit P/F
 - $N(R)$
 - $N(S)$
 - Bits de código
29. O menor *frame* no protocolo HDLC é usualmente o *frame* _____.
- de informação
 - de supervisão
 - Gerenciamento
 - Nenhuma das respostas anteriores
30. Quando dados e ACKs são enviados num mesmo *frame*, isto é denominado _____.
- Piggybacking
 - Backpacking
 - Piggypacking
 - Uma boa idéia

Exercícios

Acesso Ponto a Ponto

Em uma rede, dois dispositivos podem estar conectados através de um *link* dedicado ou *link* compartilhado. No primeiro caso, o *link* pode ser utilizado pelos dispositivos a qualquer momento. Referiremos a este tipo de acesso como **acesso ponto a ponto**. No segundo caso, o *link* é compartilhado entre os pares de dispositivos que o utilizam. Referiremos a este tipo de acesso como **acesso múltiplo**.

Um acesso múltiplo pode envolver acessos ponto a ponto. Quando dois dispositivos numa situação de acesso múltiplo conseguirem acesso ao *link* ou canal, pode ser que eles necessitem utilizar um protocolo de acesso ponto a ponto para troca de dados. Trataremos o acesso ponto a ponto neste capítulo e deixaremos para o Capítulo 13 o acesso múltiplo.

12.1 PROTOCOLO PONTO A PONTO

Um dos protocolos mais difundidos para acesso ponto a ponto é o **Protocolo Ponto a Ponto – PPP**. Hoje em dia, milhares de usuários da Internet conectam-se aos provedores de Internet (ISP) usando o protocolo PPP. A maioria desses usuários tem um *modem* padrão, um *modem* DSL ou um *cable modem*. Eles estão conectados à Internet através da linha telefônica ou através da conexão de TV a cabo. Via linha telefônica ou conexão de TV a cabo, que provêem a conexão física (*link* físico), o controle e o gerenciamento da transferência de dados é feito mediante a utilização de um protocolo ponto a ponto. O protocolo PPP é de longe o protocolo mais comum para essas aplicações.

O protocolo PPP proporciona muitos serviços que examinaremos neste capítulo. Ele define:

1. O formato do *frame* a ser trocado entre dispositivos.
2. Como os dispositivos podem negociar o estabelecimento e a troca de dados no *link*.
3. Como os dados da camada de rede são encapsulados em *frames* na camada de enlace.
4. De que forma dois dispositivos podem autenticar-se mutuamente.

Formato do *Frame*

O protocolo PPP incorpora uma versão do HDLC. A Figura 12.1 mostra o formato do *frame* PPP típico. A descrição de cada um dos campos é dada a seguir:

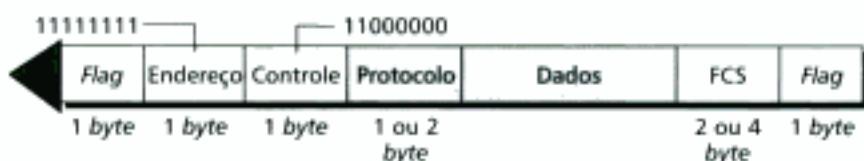


Figura 12.1 Frame PPP.

- **Campo flag.** Os campos *flag* identificam as fronteiras ou limites do *frame* PPP, seguindo a mesma linha do *frame* HDLC. O valor dos *flags* de início e fim é 01111110.
- **Campo de endereço.** Visto que o protocolo PPP é utilizado para estabelecimento de conexões ponto a ponto, ele utiliza o endereço de *broadcast* do HDLC (11111111) para evitar o endereço do enlace de dados no protocolo.
- **Campo controle.** O campo controle utiliza o formato do U-frame do protocolo HDLC. O valor padrão é 11000000 para mostrar que o *frame* não contém nenhuma seqüência numérica e que não existem mecanismos de controle de fluxo ou erro.
- **Campo protocolo.** O campo protocolo define o que efetivamente está sendo transportado no campo de dados: dados em si ou outro tipo de informação. Analisaremos esse campo em detalhes nas próximas seções.
- **Campo de dados.** Este campo pode conter dados ou outro tipo de informação. Voltaremos ao campo de dados mais adiante neste capítulo.
- **Campo FCS.** Assim como no HDLC, o campo FCS do PPP é simplesmente um CRC de 2 ou 4-bytes.

Transição de Estados

Uma conexão PPP passa por diferentes etapas as quais estão ilustradas no diagrama de **transição de estados** (veja a Figura 12.2).

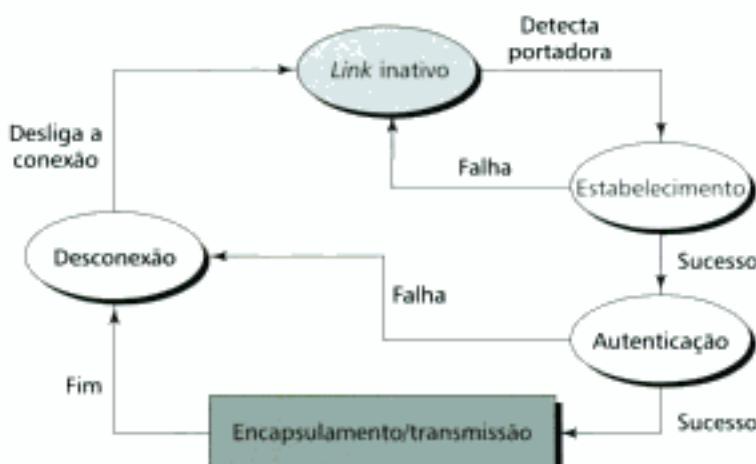


Figura 12.2 Transição de estados.

- **Idle state (ocioso ou inativo).** O *idle state* representa a inatividade do *link*, isto é, não há nenhuma portadora ativa e a linha permanece em silêncio.
- **Establishing state (estabelecimento).** Quando uma das extremidades iniciar a comunicação, a conexão é colocada no *establishing state*. Neste estado, ocorre a negociação entre as duas partes das opções de conexão. Se a negociação for bem-sucedida, o sistema passa ao estado de autenticação (quando necessário) ou segue diretamente ao estado *networking*. Os pacotes que utilizam protocolos de controle do enlace, em síntese, são utilizados para esta finalidade. Muitos pacotes podem ser trocados durante este estado.

- **Authenticating state (autenticação).** O *authenticating state* é opcional. As extremidades podem decidir, durante o estado de estabelecimento, não solicitar nenhum tipo de autenticação. Contudo, se decidirem proceder com a autenticação são enviados muitos pacotes de autenticação, que discutiremos noutra seção. Se o resultado for bem sucedido, a conexão é colocada no *networking state*. Senão, a conexão segue para o *terminating state* (desconexão).
- **Networking state (encapsulamento/transmissão).** O *networking state* é o coração dos estados de transição. Quando uma conexão chega nesse estado são encapsulados os pacotes de dados e informação de controle dos usuários para troca entre as duas extremidades. A conexão permanece ativa até que uma das pontas termine a conexão.
- **Terminating state (desconexão).** No *terminating state* são trocados muitos pacotes encerrando a negociação entre as extremidades, seguido do desligamento do *link*.

12.2 PPP: PILHA DE PROTOCOLOS

Embora o PPP seja um protocolo da camada de enlace, este protocolo é composto de uma pilha de outros protocolos para estabelecer o *link*, autenticar as duas partes e encapsular dados da camada de rede. Três conjuntos de protocolos sustentam o protocolo PPP tornando-o poderoso: Link Control Protocol (LCP), protocolos de autenticação (PAP e CHAP) e Network Control Protocol (NCP). A qualquer instante, um *frame* PPP pode estar transportando dados relativos a um destes protocolos no campo dados, conforme Figura 12.3.

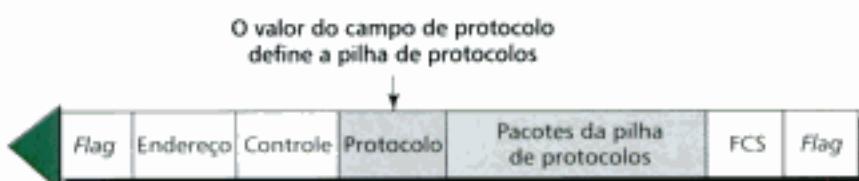


Figura 12.3 Pilha de protocolos.

Link Control Protocol (LCP)

O LCP é um dos protocolos da pilha PPP. Ele é responsável pelo estabelecimento, manutenção, configuração e terminação dos *links*. Além disso, o LCP provê mecanismos de negociação para configurar opções entre as duas extremidades. Ambas extremidades do *link* podem chegar a um acordo sobre as opções e natureza do enlace antes do *link* ser estabelecido.

Note que, quando o PPP estiver encapsulando um pacote LCP, o PPP está no *establishing state* ou *terminating state*. Nenhum conjunto de dados do usuário é encapsulado durante estes dois estados.

Todos os pacotes LCP são encapsulados no campo de dados do *frame* PPP. O que define o tipo de *frame* é o valor do campo protocolo, cujo valor padrão é C021H*. A Figura 12.4 mostra o formato do pacote LCP.

As descrições dos campos são as seguintes:

- **Código.** Este campo define o tipo de pacote LCP. Discutiremos os propósitos de tais pacotes na próxima seção.
- **ID.** Este campo mantém um valor usado como resposta a uma solicitação. Uma das estações das extremidades insere um valor neste campo, o qual será copiado no pacote resposta.

* N. de R. T.: O número C021H está escrito na representação hexadecimal. Apenas para informação, ele equivale ao decimal 49185 ou ao binário 110000000100001.

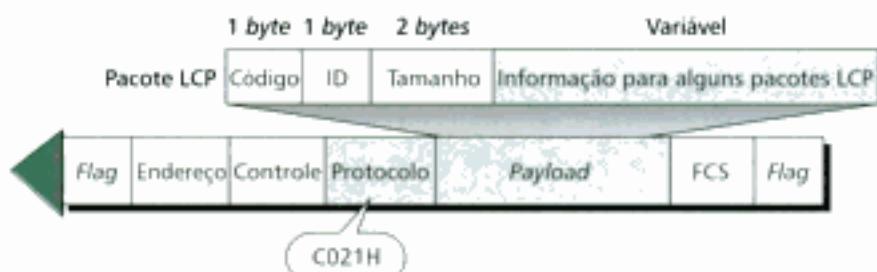


Figura 12.4 Pacote LCP encapsulado num frame PPP.

- **Tamanho.** Este campo define o tamanho integral (comprimento) do pacote LCP.
- **Informação.** Este contém informação extra necessária a alguns pacotes LCP.

Pacotes LCP

A Tabela 12.1 ilustra alguns dos pacotes LCP.

TABELA 12.1 Pacotes LCP e Descrições

Código	Tipo de pacote	Descrição
01H	Configure-request	Contém a lista de opções propostas e os respectivos valores
02H	Configure-ack	Aceita todas as opções propostas
03H	Configure-nak	Comunica que algumas opções não são aceitáveis
04H	Configure-reject	Comunica que algumas opções não são reconhecidas
05H	Terminate-request	Requisita o desligamento da linha
06H	Terminate-ack	Aceita a requisição de desligamento
07H	Code-reject	Comunica um código desconhecido
08H	Protocol-reject	Comunica um protocolo desconhecido
09H	Echo-reject	Um tipo de mensagem de "hello" para verificar se a outra extremidade está ativa
0AH	Echo-reply	A resposta à mensagem de echo-request
0BH	Discard-request	Uma requisição para descartar o pacote

Pacotes de Configuração Estes pacotes são utilizados para negociar as opções entre as duas pontas do link. São utilizados quatro tipos diferentes de pacotes para esse propósito: *configure-request*, *configure-ack*, *configure-nak* e *configure-reject*.

- **Configure-request.** A estação que deseja iniciar uma conexão envia uma mensagem de *configure-request* com uma lista de zero ou mais opções para a outra extremidade. Note que todas as opções são negociadas num único pacote.
- **Configure-ack.** Se todas as opções listadas no pacote *configure-request* são aceitas pela estação receptora, o pacote *configure-ack* é enviado de volta para a estação transmissora repetindo todas as opções requisitadas.
- **Configure-nak.** Se a estação receptora do pacote *configure-request* reconhecer todas opções, mas determinar que é necessário omitir ou revisar algumas delas (por exemplo, os valores devem ser modificados), ela envia um pacote *configure-nak* à estação transmissora. Assim, o transmissor omite ou revisa as opções e reenvia um pacote *configure-request* totalmente novo.
- **Configure-reject.** Se algumas das opções não forem reconhecidas pelo receptor, ele responde com um pacote *configure-reject*, destacando as opções que não foram reconhecidas. O transmissor da requisição deve revisar a mensagem de *configure-request* e enviar outra totalmente nova.

Pacotes de Desconexão do Link Esses pacotes são utilizados para desconectar (desligar) o *link* entre duas extremidades.

- **Terminate-request.** Ambas estações podem fechar um *link* enviando um pacote *terminate-request*.
- **Terminate-ack.** A estação que receber um pacote *terminate-request* deve responder com um pacote *terminate-ack*.

Monitoramento do Link e Depuração de Pacotes Estes pacotes são utilizados no monitoramento e depuração do *link*.

- **Code-reject.** Se uma estação receber um pacote contendo um código desconhecido, ela envia um pacote *code-reject*.
- **Protocol-reject.** Se uma estação receber um pacote contendo um protocolo desconhecido, ela envia um pacote *protocol-reject*.
- **Echo-request.** A função deste pacote é monitorar o *link*. O propósito desse pacote é verificar se o *link* está funcionando. O transmissor espera receber um pacote *echo-reply* da estação testada.
- **Echo-reply.** Este pacote serve como resposta ao pacote *echo-request*. O campo de informação no pacote *echo-request* é duplicado exatamente e reenviado ao transmissor através do pacote *echo-reply*.
- **Discard-request.** Este é um tipo de pacote *loopback* (teste). Ele é utilizado pelo transmissor para verificar a própria condição interna. O receptor do pacote apenas o descarta.

Opções

Existem muitas opções negociáveis entre duas estações. As opções são inseridas no campo de informação dos pacotes de configuração. Listamos algumas das opções mais comuns na Tabela 12.2.

TABELA 12.2 Opções típicas

Opção	Default (padrão)
Tamanho máximo do pacote recebido	1500
Protocolo de autenticação	Nenhum
Compressão do campo de protocolo	Desabilitada
Compressão dos campos de endereço e controle	Desabilitada

Protocolos de Autenticação

A autenticação exerce um papel muito importante no protocolo PPP porque este protocolo foi desenvolvido para aplicações em linhas discadas (*dial-up lines*) nas quais é necessária a verificação da identidade do usuário. O termo **autenticação** significa validação da identidade do usuário que necessitar acessar um determinado conjunto de recursos. O PPP disponibiliza dois protocolos distintos para autenticação: Password Authentication Protocol (PAP) e Challenge Handshake Authentication Protocol (CHAP). Perceba que estes protocolos são utilizados durante o estado de autenticação. Nesse estado, não são trocados dados entre usuários, somente os pacotes correspondem à autenticação do usuário.

PAP

O **Password Authentication Protocol (PAP)** é um procedimento de autenticação simples que envolve apenas dois processos:

- O usuário que desejar acessar um sistema envia algum tipo de identificador (normalmente o *logon* do usuário) e uma senha (*password*).
- O sistema verifica a validade da identificação e a senha, rejeitando ou aceitando a conexão.

A maioria dos sistemas requerem alto nível de segurança. Por isso, o PAP não é suficiente. Se outro usuário tiver acesso ao *link*, pode facilmente capturar usuário e senha e acessar os recursos do sistema. A Figura 12.5 ilustra o conceito PAP.

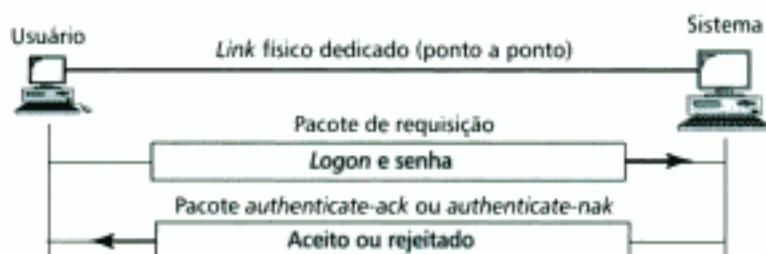


Figura 12.5 Pacotes PAP.

Pacotes PAP Os pacotes PAP são encapsulados num *frame* PAP. O que distingue um pacote PAP de outros pacotes é o valor do campo de protocolo (C023H). São três os pacotes PAP: *authenticate-request*, *authenticate-ack* e *authenticate-nak*. O primeiro pacote é utilizado pelo usuário para enviar identificação e senha; o segundo é utilizado pelo sistema para permitir o acesso e o terceiro é usado pelo sistema para negar o acesso. A Figura 12.6 mostra o formato dos três pacotes.

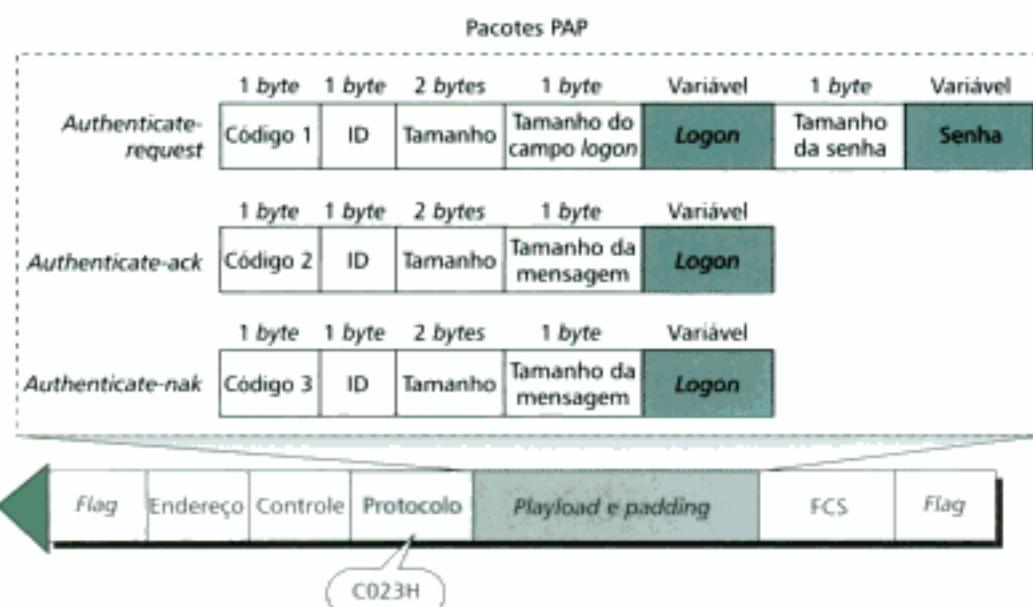


Figura 12.6 Pacotes PAP.

CHAP

O **Challenge Handshake Authentication Protocol (CHAP)** é um protocolo de autenticação que utiliza o algoritmo *three-way handshaking** (*handshaking triplo*) para agregar um nível de segu-

* N. de R. T.: *Handshaking* é uma sequência de mensagens trocadas entre dois ou mais dispositivos de rede para garantir a sincronização da transmissão.

rança à conexão superior ao método PAP. Neste método, a senha é mantida em segredo e nunca é enviada através da linha.

- O sistema envia ao usuário um pacote de requisição (*challenge request*) contendo o valor da requisição, usualmente em poucos bytes.
- O usuário aplica uma função predeterminada que toma o valor da requisição e a própria senha para criar um resultado. O usuário envia o resultado no pacote resposta (*response*) ao sistema.
- O sistema faz a mesma operação. Ele aplica a mesma função à senha do usuário (conhecida pelo sistema) e ao valor da requisição para produzir um resultado. O acesso é permitido se, e somente se, o resultado gerado pelo sistema coincidir com o valor do pacote resposta.
- Especialmente nos sistemas onde o valor da requisição (*challenge*) é modificado continuamente, o método CHAP é muito mais seguro que o PAP. A senha permanecerá secreta até mesmo se um intruso descobrir o valor da requisição (*challenge*) e do resultado. A Figura 12.7 ilustra o conceito CHAP.

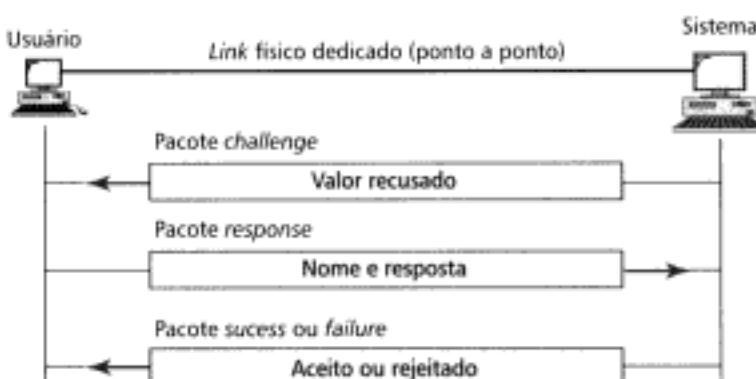


Figura 12.7 CHAP.

Pacotes CHAP

Os pacotes CHAP são encapsulados no *frame PPP*. O que distingue um pacote CHAP dos demais é o valor do campo protocolo (C223H). Temos ao todo quatro pacotes CHAP, são eles: *challenge*, *response*, *success* e *failure*. Como vimos, o pacote *challenge* é utilizado pelo sistema para enviar o valor da requisição. O pacote *response* é utilizado pelo usuário para retornar o resultado do cálculo. Já o pacote *success* é a confirmação que libera o acesso ao sistema e, por fim, o pacote *failure* é utilizado quando o acesso ao sistema é negado. A Figura 12.8 ilustra o formato padrão dos quatro tipos de pacotes.

Network Control Protocol (NCP)

Após o estabelecimento do *link* e o pedido de autenticação (se houver) ser bem sucedido, a conexão avança para o *networking state* (encapsulamento/transmissão). Neste estado, o protocolo PPP chama o **Network Control Protocol (NCP)**. O NCP é um conjunto de protocolos que permitem o encapsulamento dos dados dos protocolos da camada de rede no *frame PPP*.

IPCP

O PPP requer dois protocolos de negociação para acessar não somente a camada de enlace, mas também a camada de rede. Antes dos dados serem enviados é estabelecida uma conexão neste nível. O conjunto de pacotes que estabelece e finaliza uma conexão de pacotes IP no nível de rede (veja Capítulo 19) é denominada **Internetwork Protocol Control Protocol (IPCP)**. O formato de um pacote IPCP é mostrado na Figura 12.9. Perceba que o valor do campo identificador de protocolo é 8021H, que define encapsulamento do pacote IPCP no *frame PPP*.

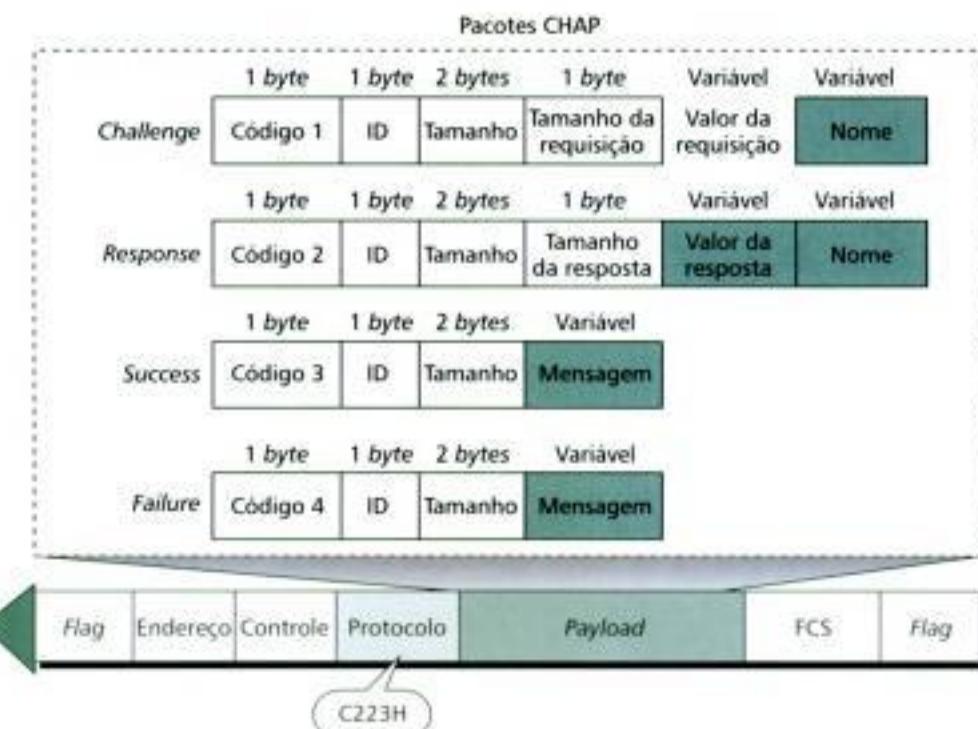


Figura 12.8 Pacotes CHAP.

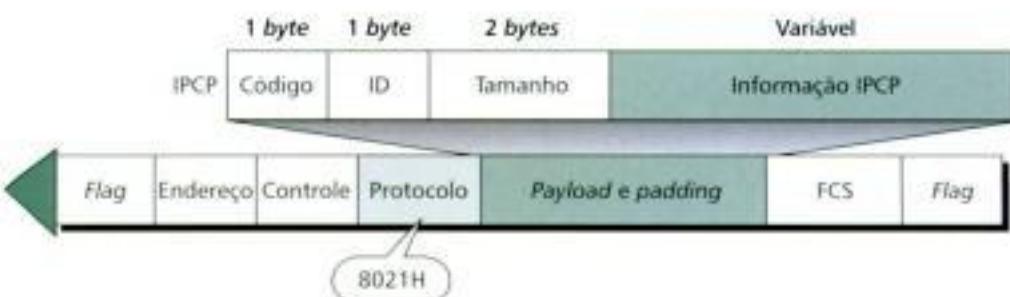


Figura 12.9 Pacote IPCP encapsulado no frame PPP.

Estão definidos sete pacotes para o protocolo IPCP, como mostrado na Tabela 12.3. O campo código é o que distingue os tipos de pacotes:

TABELA 12.3 Códigos para Pacotes IPCP

Código	Pacote IPCP
01	Configure-request
02	Configure-ack
03	Configure-nak
04	Configure-reject
05	Terminate-request
06	Terminate-ack
07	Code-reject

Uma estação utiliza o pacote *configure-request* para negociar opções com a outra estação do link, para configurar os endereços IP e assim por diante.

Passado o estado de configuração, o *link* está pronto para transportar dados IP no campo *payload* do *frame* PPP. Desta vez, o valor do campo identificador de protocolo é 0021H, para mostrar que um pacote de dados IP e não um pacote IPCP, está sendo transportado no *link*.

Terminado todos os pacotes IP, o IPCP pode tomar o controle e usar os pacotes de *terminate-request* e *terminate-ack* para finalizar o enlace da rede.

Outros Protocolos

Embora nossa discussão tenha ficado limitada ao uso dos pacotes de Internet, o PPP pode encapsular diferentes tipos de pacotes de camada superior.

Um Exemplo

Vamos examinar os estados pelos quais os pacotes da camada de rede são encapsulados e transmitidos através da conexão PPP. A Figura 12.10 ilustra os passos:

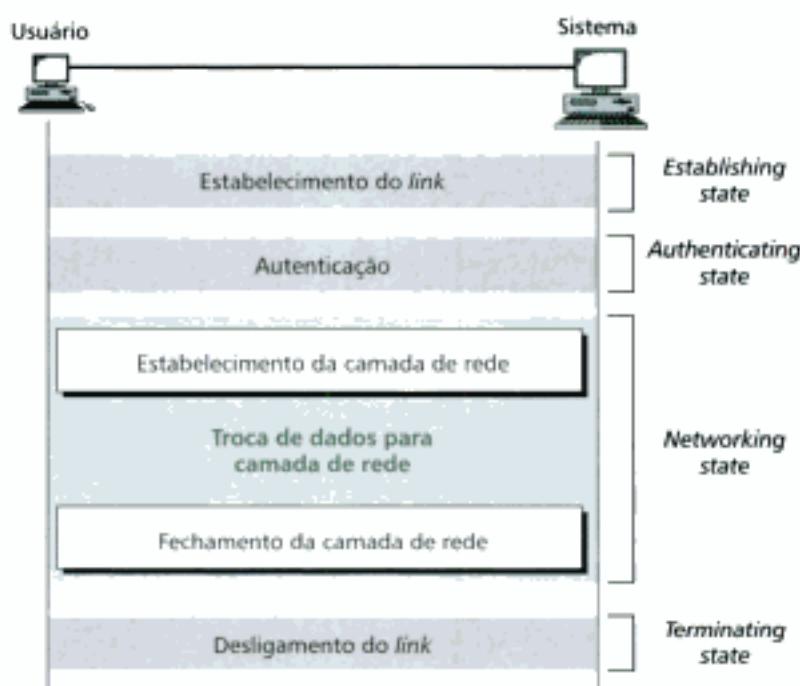


Figura 12.10 Um exemplo.

- **Establishing**. O usuário envia um pacote *configure-request* para negociar as opções de acesso e estabelecer o enlace (*link*). O usuário solicita autenticação PAP. Recebido o pacote *configure-ack*, o *link* é estabelecido.
- **Authenticating**. O usuário envia um pacote *authenticate-request* o qual inclui o logon e senha do usuário. Recebido o pacote *configure-ack*, o estado de autenticação é finalizado.
- **Networking**. Neste estado, o usuário envia um pacote *configure-request* para negociar as opções de encapsulamento dos dados da camada de rede. Recebido um *configure-ack*, o usuário pode transmitir os dados da camada de rede*, os quais são constituídos de múltiplos *frames*. Terminada a transmissão de dados, o usuário transmite um pacote *termina-*

* N. de R. T.: A rigor, os dados da camada de rede (cabeçalho + dados + checksum) são encapsulados como datagramas na camada de enlace. Se a pilha de protocolos de camada superior for o TCP/IP, os datagramas IP serão encapsulados na camada de enlace e transportados pelos *frames* PPP.

te-request para finalizar o *link*. Recebido o pacote *terminate-ack*, o estado de encapsulamento/transmissão está completo. A conexão é colocada no *terminating state*.

- **Terminating.** O usuário envia o pacote de *terminating-request* e desconecta o *link*. O *link* é finalizado tão logo o usuário receba o pacote *terminating-ack*.

12.3 TERMOS-CHAVE

Acesso ponto a ponto	Link Control Protocol (LCP)
Autenticação	Network Control Protocol (NCP)
<i>Authenticating state</i> (estado de autenticação)	<i>Networking state</i> (estado de encapsulamento/transmissão)
Challenge Handshake Authentication Protocol (CHAP)	Password Authentication Protocol (PAP)
Diagrama de estados de transição	Protocolo Ponto a Ponto (PPP)
<i>Establishing state</i> (estado de estabelecimento)	<i>Terminating state</i> (estado de desconexão ou finalização)
<i>Idle state</i> (estado ocioso ou inativo)	
Internetwork Protocol Control Protocol (IPCP)	

12.4 RESUMO

- O Protocolo ponto a ponto (PPP) é um protocolo desenvolvido para disponibilizar aos usuários acesso à Internet através de linhas dedicadas, via linha telefônica ou conexão de TV a cabo.
- Uma conexão PPP passa pelos seguintes estados: *idle*, *establishing*, *authenticating* (opcional), *networking* e *terminating*.
- O PPP utiliza uma versão do protocolo de camada de enlace HDLC.
- O Link Control Protocol (LCP) é responsável pelo estabelecimento, manutenção, configuração e terminação dos enlaces (*links*).
- Password Authentication Protocol (PAP) e Challenge Handshake Authentication Protocol (CHAP) são dois protocolos de autenticação utilizados pelo PPP.
- PAP é um processo de dois passos. O usuário transmite a identificação (normalmente o *logon*) e a senha. O sistema determina a autenticidade da informação recebida.
- CHAP é um processo de três passos. O sistema envia uma requisição numérica (o *challenge*) ao usuário. O usuário manipula a requisição e transmite o resultado. O sistema verifica o resultado.
- Network Control Protocol (NCP) é um conjunto de protocolos que permite o encapsulamento dos dados dos protocolos de camada de rede. Cada conjunto é específico do protocolo da camada de rede que solicita os serviços PPP.
- Internetwork Protocol Control Protocol (IPCP), um protocolo NCP, estabelece e finaliza uma conexão de pacotes de camada de rede (IP).

12.5 PRATIQUE OS CONHECIMENTOS ADQUIRIDOS

Questões de Revisão

1. Que tipo de usuário necessita do protocolo PPP?
2. Descreva cada um dos estados de uma conexão PPP.
3. Cite os três protocolos que constituem a pilha PPP.
4. Qual é a finalidade do campo protocolo num *frame* PPP?
5. Descreva, sucintamente, o campo controle do *frame* PPP.
6. Qual é a finalidade do LCP?
7. Estabeleça a relação entre o pacote LCP e o *frame* PPP.
8. Quais são os tipos de pacotes LCP? Qual é a função de cada um?
9. Quais os dois tipos de protocolos de autenticação no PPP?
10. Como funciona o PAP? Qual é a maior deficiência desse protocolo?
11. Como funciona o CHAP? Por que ele é superior ao PAP?
12. De que maneira um *frame* PPP transporta pacotes de autenticação do PAP e do CHAP?
13. Qual é a finalidade do NCP?
14. Estabeleça a relação entre IPCP e NCP.

Questões de Múltipla Escolha

15. De acordo com o diagrama de estados de transição do protocolo PPP, trocas de pacotes de dados e informação de controle do usuário ocorrem no _____ state.
- Establishing*
 - Authenticating*
 - Networking*
 - Terminating*
16. De acordo com o diagrama de estados de transição do protocolo PPP, as opções são negociadas no _____ state.
- Establishing*
 - Authenticating*
 - Networking*
 - Terminating*
17. De acordo com o diagrama de estados de transição do protocolo PPP, verificação da identificação do usuário ocorre no _____ state.
- Establishing*
 - Authenticating*
 - Networking*
 - Terminating*
18. De acordo com o diagrama de estados de transição do protocolo PPP, o link é desconectado no _____ state.
- Establishing*
 - Authenticating*
 - Networking*
 - Terminating*
19. No frame PPP, o campo _____ define o conteúdo do campo de dados.
- Flag*
 - Controle*
 - Protocolo*
 - FCS*
20. No frame PPP, o campo _____ define o conteúdo do campo de dados.
- Flag*
 - Controle*
 - Protocolo*
 - FCS*
21. No frame PPP, o campo _____ possui o valor 11111111 para indicar o endereço de broadcast do HDLC.
- Flag*
 - Controle*
22. No frame PPP, o campo _____ é para controle de erro.
- Flag*
 - Controle*
 - Protocolo*
 - FCS*
23. Qual é a finalidade dos pacotes LCP?
- Configuração
 - Terminação
 - Negociação das opções
 - Todas acima
24. _____ é um protocolo *three-way handshake* para verificação do usuário.
- PPP
 - CHAP
 - PAP
 - (b) e (c)
25. Os pacotes PAP e CHAP podem ser distinguídos através do valor campo _____ do frame PPP.
- Endereço
 - Controle
 - Protocolo
 - FCS
26. PAP requer _____ e _____ do usuário.
- Uma senha; um valor calculado
 - Um *logon*; uma senha
 - Um valor (*challenge*); uma senha
 - Um *logon*; um valor calculado
27. Para autenticação CHAP, o usuário recebe _____ do sistema e gera a própria _____ para criar um resultado que é então enviado ao sistema.
- Um *logon*; senha
 - Uma senha; *challenge*
 - Uma senha; identificação (*logon*)
 - Um valor (*challenge*); senha
28. _____, um protocolo _____, estabelece e termina uma conexão entre camadas de rede para pacotes IP.
- NCP; IPCP
 - CHAP; NCP
 - IPCP; NCP
 - SLIP; PPP

Exercícios

29. Quais são os valores dos campos *flag*, endereço e controle em hexadecimal?
30. Construa uma tabela para comparar o *frame PPP* com o *U-frame HDLC*. Quais campos são idênticos? Quais são diferentes?
31. O valor dos primeiros *bytes* de um *frame* é 7EFFC0C02105H. Que protocolo é encapsulado no campo *payload*? Qual é o tipo de pacote?
32. O valor dos primeiros *bytes* de um *frame* é 7EFFC0C02109110014H. Que protocolo é encapsulado no campo *payload*? Que tipo de pacote está sendo transportado? Quantos *bytes* de informação há no pacote?
33. Apresente o conteúdo de um pacote *configure-nak* no LCP. Encapsule o pacote num *frame PPP*.
34. Apresente o conteúdo de um pacote *configure-nak* no NCP. Encapsule o pacote num *frame PPP*.
35. Compare os resultados dos Exercícios 33 e 34. Que diferença você vê?
36. Apresente o conteúdo de um pacote *echo-request* com a mensagem "hello". Escreva todo o pacote em hexadecimal. Encapsule o pacote num *frame PPP* e mostre o resultado em hexadecimal.
37. Apresente o conteúdo do pacote resposta (*echo-reply*) do Exercício 36. Escreva todo o pacote em hexadecimal. Encapsule o pacote num *frame PPP* e mostre o resultado em hexadecimal.
38. Apresente o conteúdo do pacote *authenticate-request* usando "Forouzan" como *logon* e "797979" como senha. Encapsule o pacote num *frame PPP*.
39. Apresente o conteúdo do pacote *authenticate-ack* recebido em resposta ao pacote do Exercício 38.
40. Apresente o conteúdo do pacote *challenge* (CHAP) usando A4253616H como valor da requisição (*challenge*). Encapsule o pacote num *frame PPP*.
41. Mostre o conteúdo do pacote *response* (CHAP) usando 6163524AH como valor resposta. Encapsule o pacote num *frame PPP*.
42. Um sistema envia o *challenge* 2A2B1425H. A senha do usuário é 22112211H. A função a ser utilizada pelo usuário adiciona os valores do *challenge* e da senha. O resultado deve ser dividido em dois e ter as ordens permutadas para obter a resposta. Apresente a resposta do usuário.
43. Se um usuário envia um pacote LCP contendo o código 02H, para que estado de transição a conexão irá após este evento?
44. Uma conexão encontra-se no *establishing state*. Se o usuário recebe um pacote LCP *configure-nak*, qual será o novo estado?
45. Uma conexão encontra-se no *networking state*. Se o usuário recebe um pacote NCP *configure-nak*, qual será o novo estado?
46. Apresente o conteúdo de todos os *frames* da Figura 12.10. Que protocolo (LCP, NCP, autenticação e assim por diante) está envolvido em cada transmissão?

Acesso Múltiplo

Quando nós de redes ou estações são conectados de maneira a compartilhar um *link* em comum, denominado *link multiponto* ou *broadcast*, precisamos de um protocolo de acesso múltiplo para coordenar o acesso ao *link*. O problema de controlar o acesso ao meio é similar às regras de conversação numa reunião. Procedimentos diferentes devem garantir o direito à fala, evitando que duas pessoas falem ao mesmo tempo, interrompam-se mutuamente, monopolizem a discussão e assim por diante.

O cenário é o mesmo nas redes multiponto. Muitos protocolos foram desenvolvidos formalmente para controlar o acesso ao *link* compartilhado. Classificamos esses protocolos em três grupos. Os protocolos de cada grupo podem ser vistos na Figura 3.1.



Figura 13.1 Protocolos de acesso múltiplo.

13.1 ACESSO ALEATÓRIO

No método de acesso aleatório cada estação tem direito ao meio, sem ser controlada por outra estação. Entretanto, se mais de uma estação tentar transmitir, ocorre um conflito de acesso (**colisão**) e os *frames* serão destruídos ou modificados. Para evitar colisão ou para decidir o que fazer, quando uma colisão acontecer, necessitamos de um procedimento que resolva as seguintes questões:

- Quando uma estação pode acessar o meio?

- O que faz uma estação se o meio estiver ocupado?
- Como uma estação determina o sucesso ou a falha de uma transmissão?
- O que faz uma estação se ocorrer uma colisão?

Os métodos de acesso aleatório que estudaremos neste capítulo estão interligados conforme a Figura 13.2.

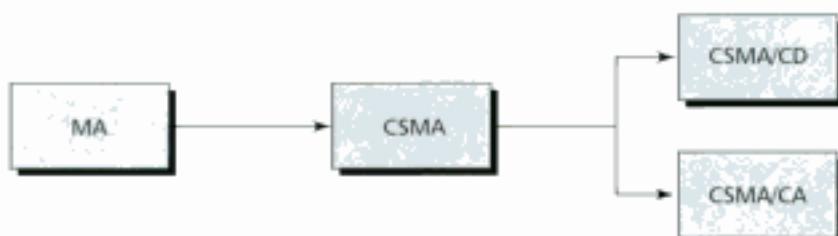


Figura 13.2 Evolução dos métodos de acesso aleatórios.

O primeiro método, conhecido como ALOHA, utiliza um procedimento muito simples denominado **acesso múltiplo (Multiple Access – MA)**. Este método foi melhorado com a adição de um procedimento que força a estação a “ouvir” o meio antes de iniciar uma transmissão. Esta técnica é denominada CSMA (Carrier Sense Multiple Access). Mais tarde, este método evoluiu em duas direções: CSMA/CD e CSMA/CA. O CSMA/CD (CSMA with Collision Detection) define procedimentos a serem seguidos se uma colisão for detectada, enquanto o CSMA/CA (CSMA with Collision Avoidance) define procedimentos para evitar uma colisão.

Acesso Múltiplo (Multiple Access – MA)

O primeiro método de acesso aleatório, batizado de **ALOHA**, foi desenvolvido no início dos anos setenta na Universidade do Havaí. Ele foi projetado para ser utilizado via ondas de rádio (*wireless*), formando uma WLAN capaz de transmitir dados a uma taxa máxima de 9600 bps.

A Figura 13.3 mostra a idéia básica por detrás da rede ALOHA. Uma estação base é eleita controladora central. Toda estação que necessitar enviar um *frame* para outra estação deve, primeiramente, se reportar à estação base. A estação base recebe o *frame* e o retransmite à estação destino. Noutras palavras, a estação base age como um salto (*hop*). A transmissão de dados na direção de *uploading* (da estação transmissora à estação base) usa modulação com uma portadora de frequência de 407 MHz. A transmissão de dados na direção de *downloading* (da estação base à estação destino) utiliza portadora de frequência 413 MHz.

A possibilidade de existir colisão neste arranjo é potencialmente grande. O meio (ar) é compartilhado entre as estações. No instante em que uma estação iniciar uma transmissão de dados para a estação base, outra estação também pode estar iniciando uma transmissão no mesmo sentido. Nesse caso, os dados de ambas estações colidem e tornam-se modificados.

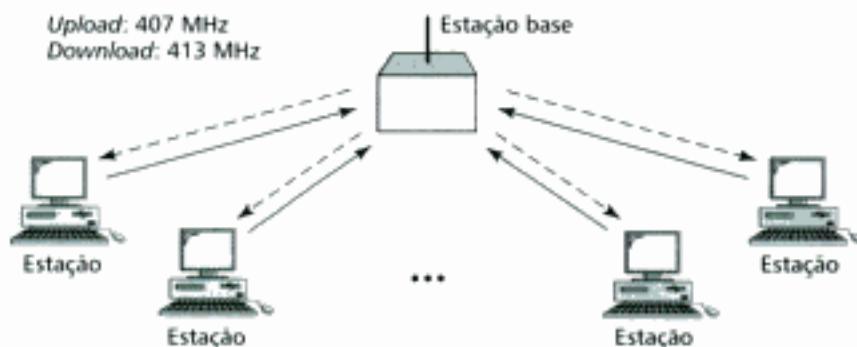


Figura 13.3 Rede ALOHA.

O protocolo ALOHA é muito simples. Ele está baseado nas seguintes regras:

- **Acesso múltiplo.** Qualquer estação pode enviar um *frame* quando quiser iniciar uma transmissão.
- **Confirmação (ACK).** Transmitido o *frame*, a estação espera por um ACK explícito ou implícito. Se ela não receber um ACK durante um certo intervalo de tempo (*slot-time*), o qual é 2 vezes o atraso de propagação máximo (o tempo gasto para o primeiro *bit* do *frame* alcançar a outra estação), a estação transmissora assume como perda do *frame*. Ela conta um tempo aleatório e tenta retransmitir os dados.

O fluxograma do protocolo é mostrado na Figura 13.4. A estação que desejar enviar um *frame* pode iniciar a transmissão sem solicitar permissão. Em seguida, ela espera por um intervalo de tempo igual a duas vezes ao atraso máximo de propagação. Se a estação receber um ACK, a transmissão foi bem sucedida. Senão, a estação utiliza uma estratégia denominada *backoff** (explicada mais adiante neste capítulo) e envia o pacote novamente. Após algumas tentativas, a estação desiste se não receber um ACK.

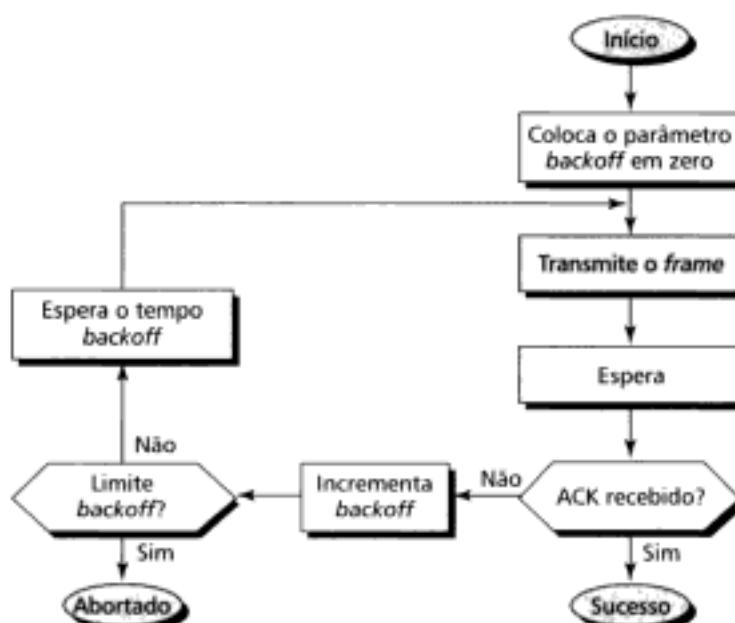


Figura 13.4 Fluxo do protocolo ALOHA.

Carrier Sense Multiple Access (CSMA)

O método de CSMA foi desenvolvido para minimizar a probabilidade de ocorrência de colisão e, portanto, melhorar a *performance* do sistema. A chance de colisão é reduzida se uma estação verificar (sentir) o meio antes de tentar utilizá-lo. No método baseado no **Carrier Sense Multiple Access (CSMA)**, primeiramente, cada estação ouve o meio (verifica o estado do meio) antes de iniciar uma transmissão. Noutras palavras, o CSMA baseia-se no princípio “verificar antes de transmitir” ou “ouvir antes de falar”.

O CSMA reduz a possibilidade de colisão, mas nunca pode reduzi-la a zero. Pode ser perguntado: por que ainda existe chance de colisão se todas estações ouvem o meio antes de iniciar a transmissão de *frames*? Essa possibilidade ainda existe devido ao atraso de propagação dos *frames*. Quando uma estação transmite um *frame*, decorre um curto intervalo de tempo antes que os primeiros *bits* desse *frame* cheguem às demais estações, para que elas possam ouvi-los. Sendo assim, uma estação que tiver dados a transmitir poderá verificar o meio e concluir que ele está ocioso, simples-

* N. de R. T.: Em síntese, *backoff* é o atraso de retransmissão imposto pelo protocolo de acesso ao meio quando ocorrer uma colisão.

mente porque o *frame* de outra estação transmissora não teve tempo hábil para se propagar até a estação que deseja iniciar uma transmissão. A Figura 13.5 ilustra como pode acontecer uma colisão no método CSMA.

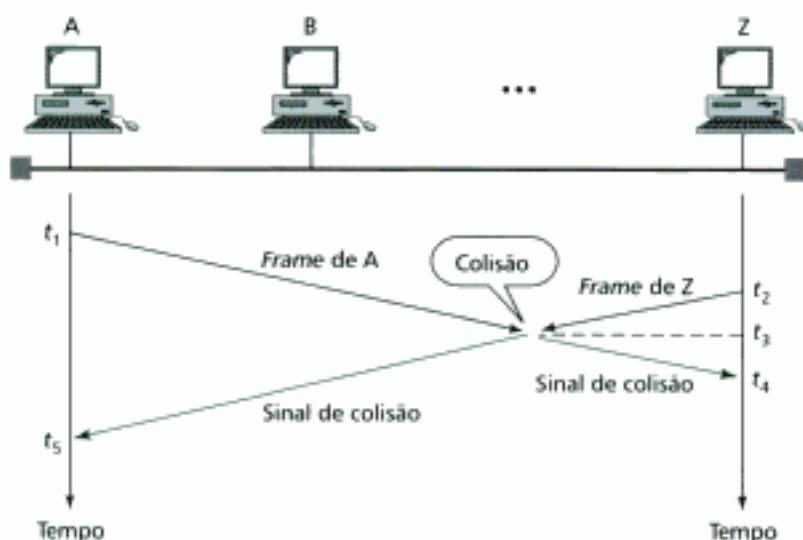


Figura 13.5 Colisão no CSMA.

Na figura anterior, a estação A verifica o meio no tempo t_1 . Ela conclui que o meio está ocioso e inicia a transmissão do *frame*. No tempo t_2 ($t_2 > t_1$), a estação Z verifica o meio e conclui que ele está disponível para transmissão (ocioso) porque, devido ao atraso de propagação, o *frame* transmitido pela estação A não teve tempo de chegar até a estação Z para que ela pudesse ouvi-lo no meio. Portanto, a estação Z transmite um *frame*. Os dois sinais colidem no tempo t_3 ($t_3 > t_2 > t_1$). Perceba que o resultado da colisão é um sinal modificado propagando-se em ambas direções. Esse sinal chega às estações Z e A, respectivamente, nos tempos t_4 ($t_4 > t_3 > t_2 > t_1$) e t_5 ($t_5 > t_4 > t_3 > t_2 > t_1$).

Estratégias de persistência

As **estratégias de persistência** definem os procedimentos para que uma estação “sinta” o meio até que ele fique livre ou disponível. Essas estratégias são: persistente e não persistente (veja a Figura 13.6).

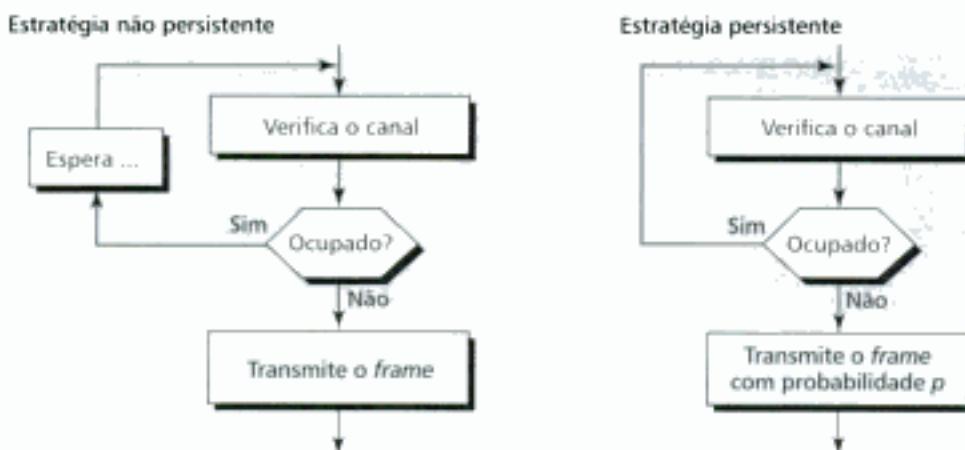


Figura 13.6 Estratégias de persistência.

Não Persistente Na **estratégia não persistente**, uma estação que tiver *frames* a transmitirouve o meio. Se o meio estiver livre, a estação inicia a transmissão imediatamente. Se o meio estiver ocupado, a estação espera um período de tempo aleatório e, então, ouve o meio novamente. A proposta

não persistente reduz a chance de colisão porque é improvável que duas ou mais estações esperem o mesmo intervalo de tempo e tentem, outra vez, transmitir simultaneamente. Entretanto, este método reduz a *performance* da rede se o meio estiver ocioso e houver estações desejando enviar *frames*.

Persistente Na estratégia **persistente**, uma estação sente o meio. Se o meio estiver livre, a estação transmite um *frame*. Este método possui duas variantes: **1-persistent** e **p-persistent**.

No método **1-persistent**, se uma estação verificar que o meio está livre, ela inicia, com probabilidade 1, a transmissão do *frame* dela. Este método aumenta a chance de colisão porque duas ou mais estações podem enviar *frames* após determinar que o meio está livre.

No método **p-persistent**, se uma estação verificar que o meio está livre, ela pode iniciar a transmissão ou não. Ela possui uma probabilidade p de transmitir e $(1 - p)$ de esperar outro instante de transmissão. Por exemplo, se $p = 0,2$ significa que cada estação, após sentir o estado do meio, envia com probabilidade 0,2 ou 20% do tempo e espera por outro instante de tempo para transmissão com probabilidade 0,8 ou 80% do tempo. A estação gera um número aleatório entre 1 e 100. Se o número escolhido aleatoriamente for menor que 20, a estação transmite. De outro modo, a estação aguarda outro momento para transmitir. A estratégia **p-persistent** combina as vantagens das duas outras estratégias. Ela reduz a chance de colisão e melhora a eficiência do sistema.

CSMA/CD

O método CSMA não define um procedimento para uma colisão. Esta é a razão da técnica CSMA nunca ser implementada efetivamente. A variante **Carrier Sense Multiple Access with Collision Detection (CSMA/CD)** agrupa um procedimento que trata uma colisão.

Neste método, qualquer estação pode transmitir um *frame*. Então, a estação monitora continuamente o meio para sentir se a transmissão foi bem sucedida. Caso seja, a estação deixa de monitorar o meio. Entretanto, se ocorrer uma colisão, os *frames* serão retransmitidos novamente. Para reduzir a probabilidade de colisão uma segunda vez, a estação espera um tempo, denominado **back-off**. A questão é: quanto tempo? É razoável que uma estação espere um tempo menor após a primeira colisão, aumente o tempo de espera, se ocorrer uma segunda colisão e assim por diante.

No método de espera exponencial (*exponential backoff method*), a estação espera durante um intervalo de tempo entre 0 e $2^N \times$ tempo de propagação máximo (tempo de propagação entre as duas estações mais distantes da rede), onde N é o número de tentativas de transmissão. Noutras palavras, a estação irá esperar entre 0 e $[2 \times (\text{tempo de propagação máximo})]$ após a primeira colisão, entre 0 e $[2^2 \times (\text{tempo de propagação máximo})]$ após a segunda colisão e assim por diante. A Figura 13.7 mostra o fluxo do procedimento CSMA/CD.

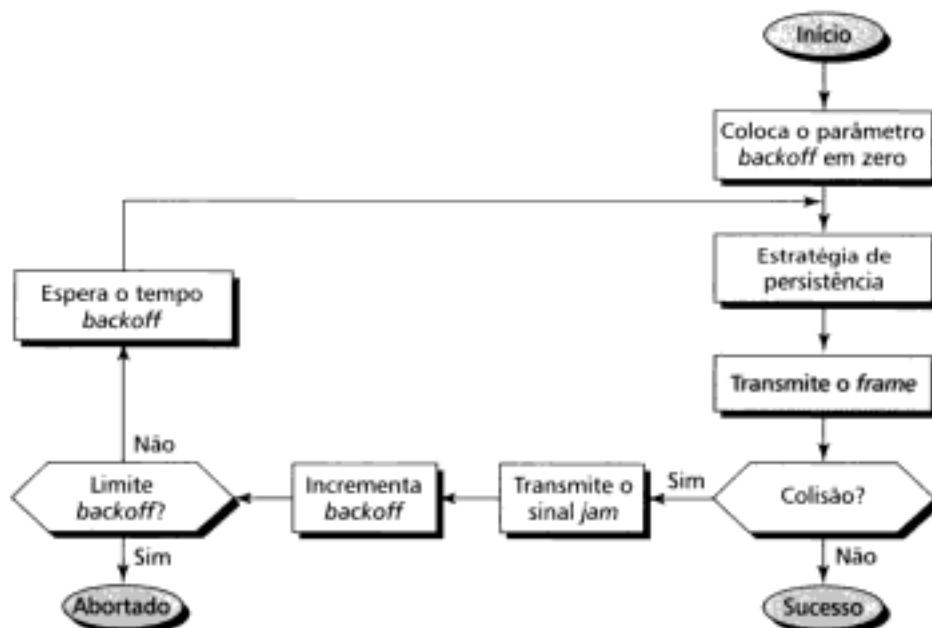


Figura 13.7 Procedimento CSMA/CD.

A estação que tiver um ou mais *frames* a transmitir posiciona o parâmetro *backoff* em zero ($N = 0$). Em seguida, ela ouve o meio usando uma das estratégias de persistência. Após transmitir o *frame*, se a estação não detectar uma colisão até que todo o *frame* tenha sido transmitido, a transmissão foi bem-sucedida. Entretanto, se a estação detectar uma colisão, ela envia um sinal de congestionamento (*jam signal*) para o meio de modo a informar às outras estações sobre a situação atual e alertá-las sobre a colisão. Todas as estações descartam o *frame* recebido. Assim, a estação transmissora posiciona o parâmetro de *backoff* para 1 ($N = 1$). Em seguida, ela testa o parâmetro N para verificar se este excede o valor limite (tipicamente 15). Se o valor exceder o limite, significa que a estação tentou o suficiente e deve desistir de tentar. A estação aborta o procedimento. Se o valor do parâmetro não excede o limite, a estação transmissora espera um tempo aleatório, baseado no parâmetro *backoff*, e volta a sentir o meio. O método CSMA/CD é utilizado nas redes Ethernet padrão (discutidas no Capítulo 14).

CSMA/CA

O procedimento CSMA/CA difere do procedimento anterior pelo fato de não haver colisão. Este procedimento procura evitar colisões (veja Figura 13.8). A estação utiliza uma das estratégias de persistência para verificar (ouvir) o meio. Se o meio estiver livre, a estação espera durante um intervalo de tempo denominado IFG (*Interframe Gap*). Então, a estação transmissora aguarda uma outra quantidade aleatória de tempo. Em seguida, transmite o *frame* e reseta o relógio (*timer*). A estação aguarda um ACK do receptor. A transmissão será bem sucedida se a estação transmissora receber um ACK antes do relógio expirar. Se a estação não receber um ACK saberá que algo está errado (*o frame* ou ACK foi perdido). A estação incrementa o valor do parâmetro *backoff*, espera um intervalo de tempo aleatório e volta a ouvir o meio. O método CSMA/CA é utilizado nas wireless LANs (veja Capítulo 15).

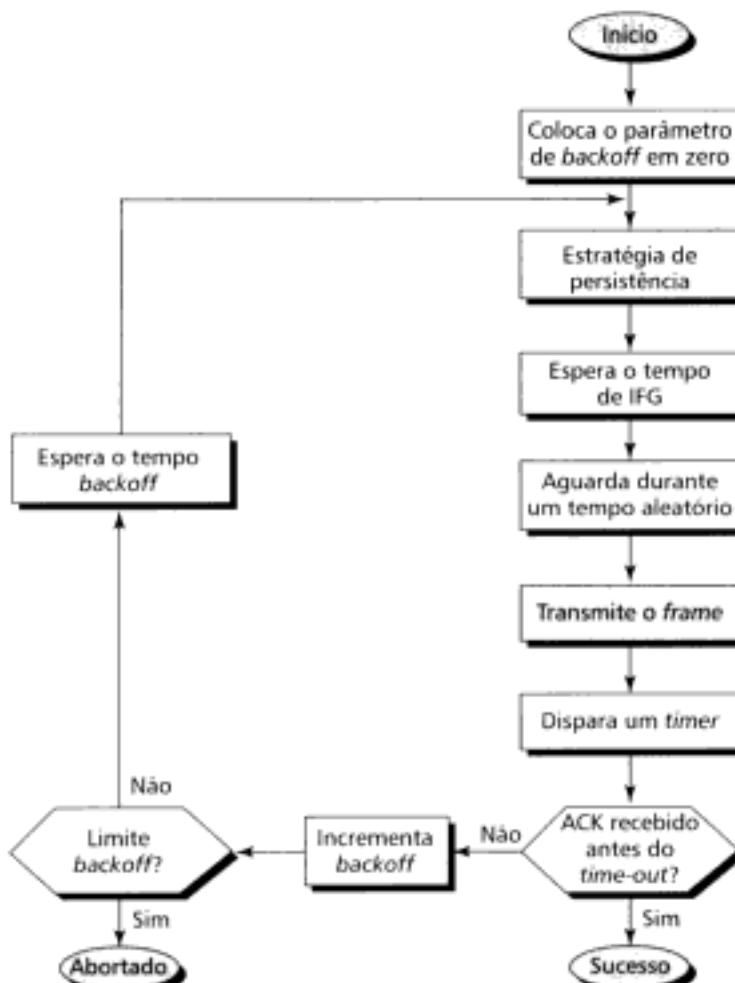


Figura 13.8 Procedimento CSMA/CA.

13.2 ACESSO ORDENADO

As estações consultam-se mutuamente para determinar qual delas terá o direito de transmitir no método de **acesso ordenado**. Uma estação não pode iniciar uma transmissão sem antes ter sido autorizada pelas demais. Examinaremos três métodos de acesso controlado bastante conhecidos.

Acesso com Reserva

Neste método de acesso, uma estação deve solicitar reserva antes de transmitir dados. O tempo é dividido em intervalos. Em cada intervalo, é enviado um *frame* de reserva antecedendo o *frame* de dados.

Se o sistema for constituído de N estações, existem exatamente N *minislots* reservados no *frame* de reserva. Cada *minislot* identifica uma estação. Quando uma estação tem *frame(s)* de dados a transmitir, faz uma reserva no *minislot* dedicado a ela. As estações que tiverem feito reserva podem enviar os *frames* de dados após a passagem do *frame* de reserva.

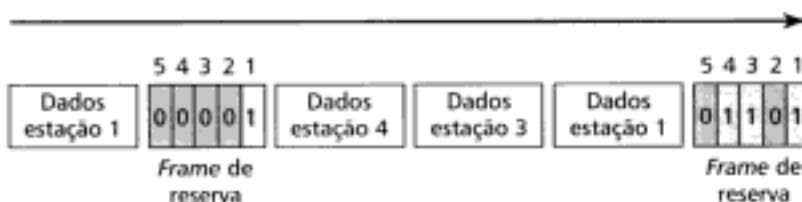


Figura 13.9 Método de acesso com reserva.

Polling

O método *polling* opera em topologias nas quais um dispositivo é eleito como **dispositivo principal** e os outros são os **dispositivos secundários**. Toda troca de dados deve passar pelo dispositivo principal, até mesmo quando o destino final é um dispositivo secundário. O dispositivo principal controla o *link* e os dispositivos secundários seguem as instruções dele. O dispositivo de rede principal indaga, de um modo ordenado, se os secundários têm dados para transmitir. A indagação ocorre na forma de uma mensagem para cada secundário que fornece, a um dispositivo secundário, o direito de transmitir durante um intervalo de tempo. Esta função é denominada **polling**. Sendo assim, sempre é o dispositivo principal quem inicia uma sessão. Se o dispositivo principal quiser enviar dados, ele pergunta ao dispositivo secundário alvo se está pronto para receber. Esta função é denominada **selecting**.

Seleção

O **modo de seleção (selecting)** é utilizado sempre que o dispositivo principal tem alguma coisa para enviar. Lembrando que é o dispositivo principal quem controla o *link*, se o dispositivo principal não estiver transmitindo nem recebendo dados, sabe que o *link* está livre (disponível). Se tiver algo a transmitir, o dispositivo principal simplesmente envia. Entretanto, o que ele não sabe a maioria das vezes é se o dispositivo secundário destino está pronto para receber. Assim, o dispositivo principal deve alertar o secundário sobre a transmissão a ser realizada e esperar por uma confirmação do secundário (o pronto!). Antes de iniciar a transmissão, o dispositivo principal cria e transmite um *select frame* (SEL), onde um dos campos inclui o endereço do secundário destino (veja a Figura 13.10).

Poll

A função *polling* é utilizada pelo dispositivo principal para solicitar transmissões dos dispositivos secundários. A Figura 3.11 mostra a situação.

Na ocasião em que o principal estiver pronto para receber dados, ele deve indagar (*poll*) cada dispositivo em volta à procura de dados a transmitir. Quando o primeiro secundário é abordado, responde com um *frame NAK*, se não tiver nada a transmitir, ou com dados (um *frame* de dados). Se a resposta for negativa (um *frame NAK*), o principal então sonda o próximo secundário da

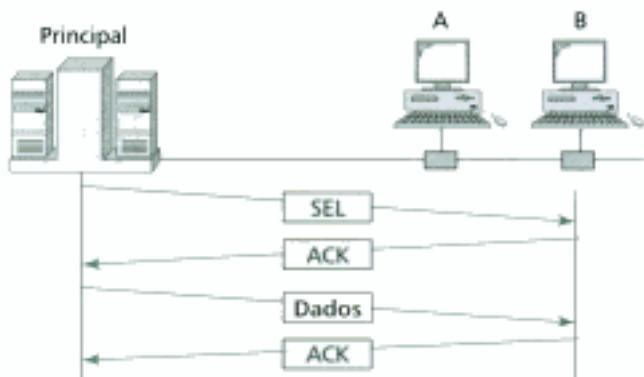


Figura 13.10 Modo de seleção (*selecting*).

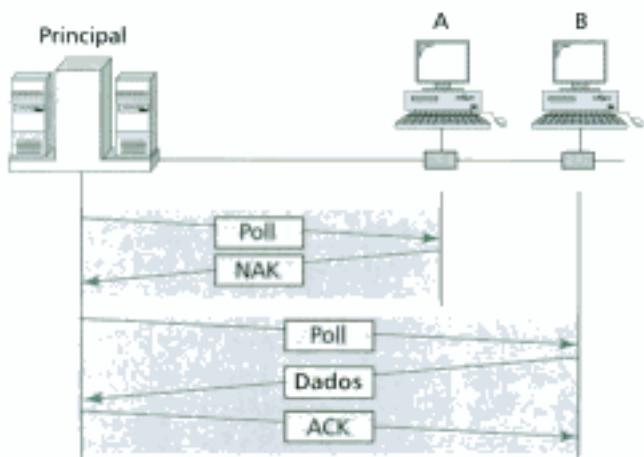


Figura 13.11 *Polling*.

mesma maneira, até encontrar um que tenha dados a transmitir. Quando a resposta for positiva (um *frame* de dados), o dispositivo principal lê o *frame* e retorna uma confirmação (um *frame* ACK) ao secundário. Em seguida, verifica o conteúdo do *frame*.

Passagem de Permissão (*Token-Passing*)

No método de **passagem de permissão** (*token-passing*), uma estação é autorizada a enviar dados quando ela recebe um *frame* especial, denominado *frame* de permissão. A topologia utilizada para organizar as estações é o anel. Cada estação possui duas estações vizinhas: predecessora e sucessora. Os *frames* vêm da estação predecessora e seguem na direção da estação sucessora. A Figura 13.12 ilustra a idéia.

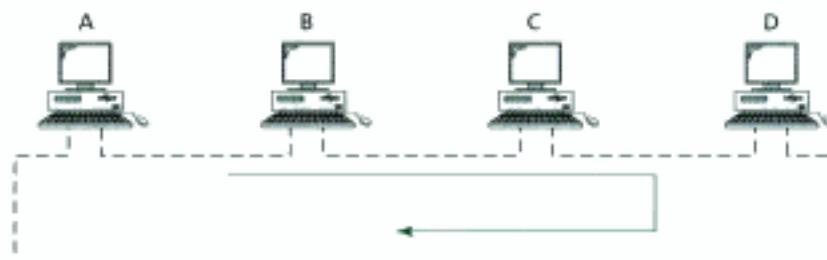


Figura 13.12 Passagem da permissão (*token-passing*).

Quando dados não estiverem sendo transmitidos, uma permissão circula pelo anel. A estação captura a permissão e a retém, enviando um ou mais *frames* (contanto que ela tenha *frames* a enviar ou o tempo disponível não tenha expirado) e, finalmente, libera a permissão para ser utilizada por outra estação.

zada pela estação sucessora (a próxima estação do anel físico ou lógico). A Figura 13.13 mostra um procedimento bastante simplificado para a passagem de permissão. Na realidade, outras características, tais como prioridade e reserva, são agregadas ao processo.

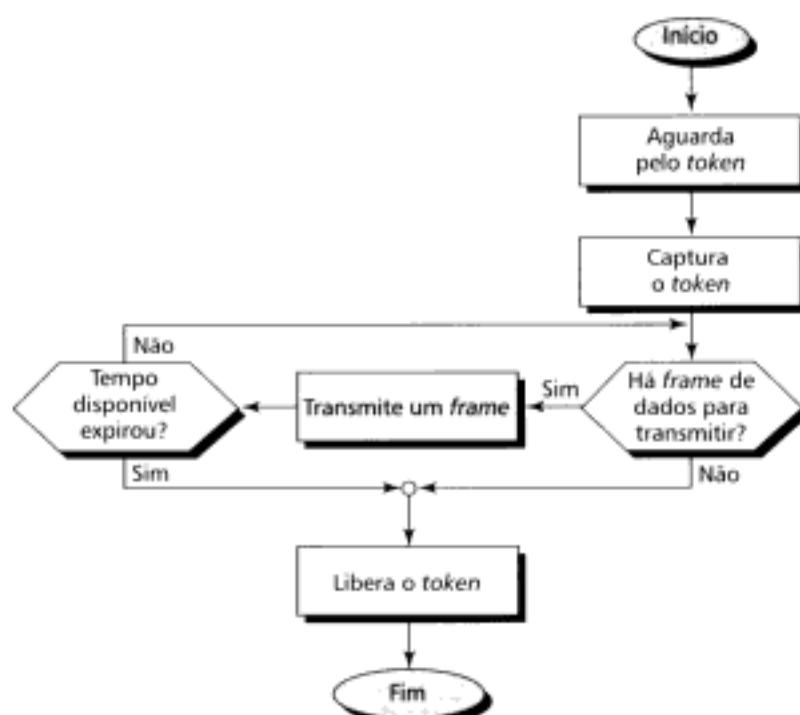


Figura 13.13 Procedimento token-passing.

13.3 CANALIZAÇÃO

A **canalização** é um método de acesso múltiplo no qual a banda disponível de um meio é compartilhada no tempo, freqüência ou através de código entre as diferentes estações de uma rede. Nesta seção, examinaremos três protocolos de canalização: FDMA, TDMA e CDMA. Os dois primeiros estão relacionados aos procedimentos discutidos previamente na camada física. O CDMA é um protocolo de acesso múltiplo ao meio.

FDMA

A largura de banda disponível no meio é compartilhada entre todas as estações no método **FDMA (Frequency-Division Multiple Access)**. Cada estação utiliza a banda que lhe é alocada para enviar e receber dados. Toda estação no sistema tem reservada uma faixa da banda disponível e, durante todo o tempo, a banda pertence exclusivamente à estação. O FDMA é um protocolo de camada de enlace que utiliza FDM na camada física (veja Capítulo 6). No Capítulo 17, veremos como o método FDMA é utilizado na telefonia celular e nas comunicações via satélite.

A largura de banda é dividida em canais ou faixas no método FDMA.

TDMA

No método **TDMA (Time-Division Multiple Access)** toda a banda pertence exclusivamente a um canal. Assim, as estações compartilham no tempo toda a capacidade do canal. A cada estação é alocado um intervalo de tempo (*slot-time*) durante o qual ela pode transmitir dados. A técnica TDMA é um protocolo de camada de enlace que utiliza TDM na camada física (veja Capítulo 6). No Capítulo 17, veremos como o TDMA é utilizado nas redes de telefonia celular.

No método TDMA, a banda é única. O canal é compartilhado no tempo.

CDMA

O método **Code-Division Multiple Access (CDMA)** foi concebido muitas décadas atrás. Os avanços recentes da tecnologia eletrônica tornaram a implementação possível. O CDMA difere do FDMA porque um único canal ocupa toda a banda no *link* e difere do TDMA porque todas as estações podem enviar dados simultaneamente, isto é, sem necessidade de compartilhar o *link* no tempo.

No método CDMA, um único canal suporta, simultaneamente, toda transmissão.

O CDMA funciona baseada na teoria da criptografia. A cada estação é atribuído um código constituído de uma seqüência de números denominados *chip* ou *bit-code*. Suponha que temos quatro estações, cada qual com a sua seqüência de *chips* designada por A, B, C e D (veja a Figura 13.14). Mais adiante, ainda neste capítulo, mostraremos como utilizar estas seqüências.

+1, +1, +1, +1	+1, -1, +1, -1	+1, +1, -1, -1	+1, -1, -1, +1
A	B	C	D

Figura 13.14 Chip code ou bit-code.

Seguiremos as seguintes regras de codificação: se uma estação precisar enviar um *bit* 0, ela enviará 0 –1. Se quiser enviar um *bit* 1, ela envia 0 +1. Uma estação não envia sinal quando ela estiver livre ou silenciosa, o qual representaremos como 0. Estas regras aparecem sintetizadas na Figura 13.15.

$$\text{bit } 0 \longrightarrow -1 \quad \text{bit } 1 \longrightarrow +1 \quad \text{Silêncio} \longrightarrow 0$$

Figura 13.15 Regras de codificação.

Analisaremos um exemplo simples para mostrar como as quatro estações compartilham o *link* durante o tempo de 1-bit. O procedimento pode facilmente ser repetido adicionando-se intervalos ao esquema. Assumiremos ainda que as estações (canais) 1 e 2 estão enviando um *bit* 0 e o canal 4 está enviando um *bit* 1. A estação 3 está em silêncio.

Multiplexador

A Figura 13.16 ilustra a situação no lado do multiplexador. As etapas são:

1. O multiplexador recebe um número codificado de cada estação (–1, –1, 0 e +1).
2. O número codificado enviado pela estação 1 é multiplicado por cada *chip* da seqüência A. O resultado é a nova seqüência (–1, –1, –1, –1). Do mesmo modo, o número codificado enviado pela estação 2 é multiplicado por cada *chip* da seqüência B. O mesmo acontece para os dois outros números das estações 3 e 4. Como resultado, obtemos quatro seqüências novas.
3. Todos os primeiros *chips* da seqüência nova são adicionados. O mesmo acontece com os segundos, terceiros e quartos *chips*. O resultado é outra seqüência nova.
4. Essa seqüência é transmitida através do *link*.

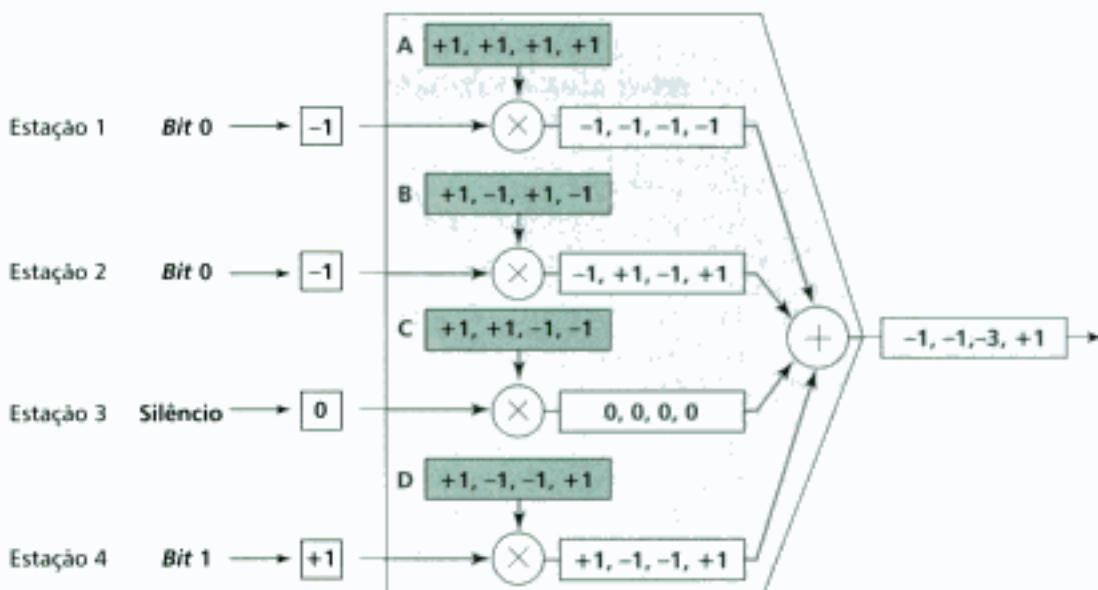


Figura 13.16 Multiplexador CDMA.

Demultiplexador

A Figura 13.17 ilustra a situação no lado do demultiplexador. As etapas são:

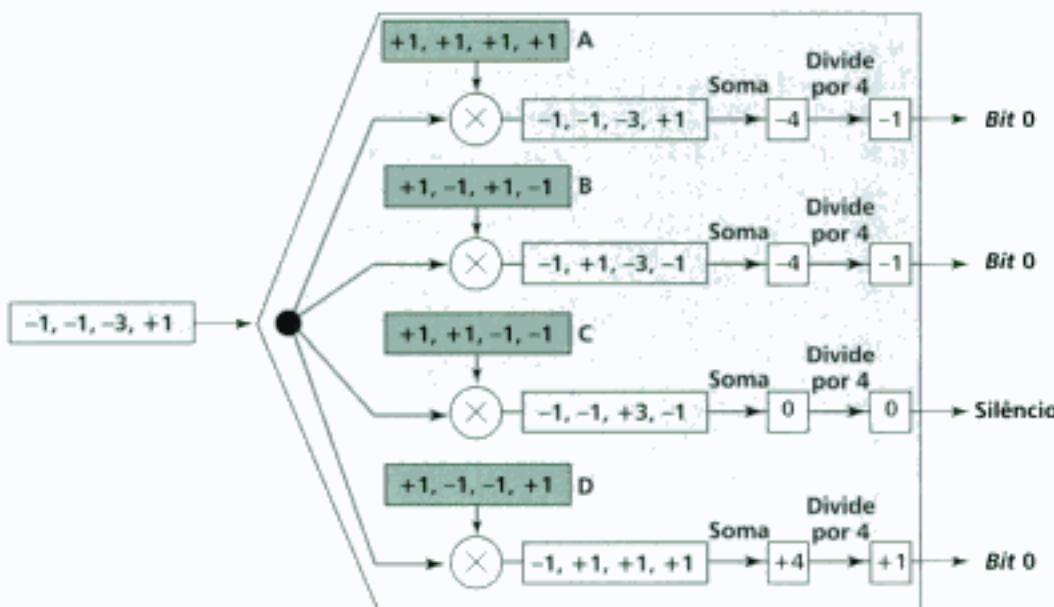


Figura 13.17 Demultiplexador CDMA.

- O demultiplexador recebe na entrada a seqüência transmitida através do link.
- Ele multiplica a seqüência pelo código de cada receptor. A multiplicação é feita *chip a chip*.
- Os *chips* em cada seqüência são adicionados. O resultado será sempre +4, -4 ou 0.
- O resultado da etapa 3 é dividido por 4 para obter -1, +1 ou 0.
- O número obtido na etapa 4 é decodificado para 0, 1 ou silêncio pelo receptor.

Observação

Através da Figura 13.17, vimos que cada estação recebe o que foi transmitido pelo canal de origem. Note que o terceiro receptor não recebe dados porque o canal equivalente no multiplexador estava

livre ou silencioso. Há somente uma única seqüência fluindo através do canal, a soma das seqüências. Entretanto, cada receptor detecta o próprio dado dentro da soma.

Seqüências Ortogonais Retornemos às seqüências de *chips*. Não escolhemos aleatoriamente tais seqüências. Elas foram escolhidas cuidadosamente. As seqüências do nosso exemplo são denominadas **seqüências ortogonais**. Mostraremos como gerar seqüências ortogonais e discutiremos as propriedades dessas seqüências.

Geração da Seqüência Para gerar seqüências usaremos a **matriz de Walsh**, uma matriz bidimensional quadrada, isto é, com mesmo número de linhas e de colunas. Cada linha contém uma seqüência de *chips*. A matriz de Walsh W_1 para uma seqüência de um único *chip* possui uma linha e uma coluna. Podemos escolher entre -1 e $+1$ para o *chip* desta matriz trivial (escolhemos $+1$). De acordo com Walsh, conhecendo a matriz para N seqüências W_N , podemos construir a matriz para $2N$ seqüências W_{2N} conforme ilustra a Figura 13.18. A seqüência W_N com a barra sobreposta representa o complemento da seqüência W_N , onde cada elemento $+1$ é substituído por -1 e vice-versa.

$$W_1 = [+1] \quad W_{2N} = \begin{bmatrix} W_N & W_N \\ \overline{W_N} & \overline{\overline{W_N}} \end{bmatrix}$$

Figura 13.18 W_1 e W_{2N}

Vamos mostrar como criar W_2 e W_4 a partir de W_1 . A Figura 13.19 ilustra o processo. Após selecionarmos W_1 , a seqüência W_2 pode ser criada de quatro W_1 , com o último elemento sendo o complemento de W_1 . Gerado o W_2 , podemos criar W_4 partindo de quatro W_2 , com a última seqüência sendo o complemento de W_2 . É claro que a seqüência W_8 é composta de quatro W_4 e assim por diante. De acordo com as matrizes, as seqüências para duas estações acessarem o link são $+1, +1$ e $+1, -1$. As seqüências para quatro estações são aquelas que utilizamos no exemplo (veja Figura 13.16).

$$W_1 = [+1] \quad W_4 = \begin{bmatrix} +1 & +1 & +1 & +1 \\ +1 & -1 & +1 & -1 \\ +1 & +1 & -1 & -1 \\ +1 & -1 & -1 & +1 \end{bmatrix}$$

$$W_2 = \begin{bmatrix} +1 & +1 \\ +1 & -1 \end{bmatrix}$$

Figura 13.19 Geração das seqüências ortogonais.

Propriedades das Seqüências Ortogonais Seqüências ortogonais têm propriedades bastante adequadas ao método CDMA. São elas:

1. Se multiplicarmos uma seqüência por -1 , cada elemento da seqüência é complementado ($+1$ torna-se -1 e vice-versa). Podemos notar que, quando uma estação está transmitindo -1 (*bit* 0), na verdade ela está transmitindo o complemento dele.
2. Se multiplicarmos dois elementos escalarmente, isto é, elemento por elemento, e adicionarmos os resultados, obteremos um número denominado **produto interno ou escalar**. Se as duas seqüências multiplicadas escalarmente forem idênticas obtemos o número N , onde N representa a quantidade de seqüências. Se as seqüências forem diferentes, o resultado do produto escalar vale 0. O produto interno utiliza um ponto como operador. Assim, $A \cdot A = N$ e $A \cdot B = 0$.
3. O produto interno de uma seqüência pelo seu complemento é $-N$. Então, $A \cdot (-A) = -N$.

Exemplo 1

Verifique se a segunda propriedade mantém-se verdadeira para a seqüência do CDMA exemplo.

Solução

O produto interno de cada seqüência do código por ela mesma deve valer N . Tomemos a seqüência do código C . Você pode verificar a veracidade da propriedade para os outros códigos.

$$C \cdot C = [+1, +1, -1, -1] \cdot [+1, +1, -1, -1] = 1 + 1 + 1 + 1 = 4$$

Se as duas seqüências forem diferentes, o produto interno vale 0.

$$B \cdot C = [+1, -1, +1, -1] \cdot [+1, +1, -1, -1] = 1 - 1 - 1 + 1 = 0$$

Exemplo 2

Verifique se a terceira propriedade sobre códigos ortogonais se mantém para a seqüência do CDMA exemplo.

Solução

O produto interno de cada código pelo complemento deve valer $-N$. Tomemos novamente a seqüência do código C . Você pode provar a veracidade da propriedade para os outros códigos.

$$C \cdot (-C) = [+1, +1, -1, -1] \cdot [-1, -1, +1, +1] = -1 - 1 - 1 - 1 = -4$$

O produto interno de um código pelo complemento de outro código vale 0.

$$B \cdot (-C) = [+1, -1, +1, -1] \cdot [-1, -1, +1, +1] = -1 + 1 + 1 - 1 = 0$$

Códigos Ortogonais em CDMA

Vamos compreender por que funcionou nosso exemplo trivial de CDMA.

No Multiplexador. Cada estação está transmitindo uma seqüência apropriada:

- A estação 1 está transmitindo $-A$ (A foi multiplicada por -1); a estação 2 está transmitindo $-B$; a estação 3 está transmitindo uma seqüência vazia (toda de zeros) e a estação 4 está transmitindo D .
- A seqüência na saída do multiplexador é a soma do resultado de todas as seqüências:

$$S = -A - B + D$$

No Demultiplexador. No demultiplexador todas as estações recebem S :

- A estação 1 faz o produto interno de S e A .

$$S \cdot A = (-A - B + D) \cdot A = -A \cdot A - B \cdot A + D \cdot A = -4 + 0 + 0 = -4$$

O resultado é então dividido por 4, o que resulta em -1 . Este é interpretado como um bit 0.

- A estação 2 faz o produto interno de S e B .

$$S \cdot B = (-A - B + D) \cdot B = -A \cdot B - B \cdot B + D \cdot B = 0 - 4 + 0 = -4$$

O resultado é então dividido por 4, o que resulta em -1 . Este é interpretado como um bit 0.

- A estação 3 faz o produto interno de S e C .

$$S \cdot C = (-A - B + D) \cdot C = -A \cdot C - B \cdot C + D \cdot C = 0 + 0 + 0 = 0$$

Este resultado é interpretado como estação em silêncio.

- A estação 4 faz o produto interno de S e D .

$$S \cdot D = (-A - B + D) \cdot D = -A \cdot D - B \cdot D + D \cdot D = 0 + 0 + 4 = 4F$$

O resultado é então dividido por 4, o que resulta em $+1$. Este é interpretado como um bit 1.

13.4 TERMOS-CHAVE

Acesso aleatório	Estação secundária
Acesso Múltiplo (MA)	Estratégia <i>1-persistent</i>
Acesso ordenado	Estratégia persistente
ALOHA	Estratégia não persistente
<i>Backoff</i>	Estratégia <i>p-persistent</i>
Canalização	Frequency-Division Multiple Access (FDMA)
Carrier Sense Multiple Access (CSMA)	Matriz de Walsh
Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA)	Modo <i>poll</i>
Carrier Sense Multiple Access/Collision Detection (CSMA/CD)	Passagem de permissão (<i>token-passing</i>)
<i>Chip</i> (bit-code)	<i>Polling</i>
Code-Division Multiple Access (CDMA)	Produto interno ou escalar
Colisão	Reserva
Estação principal	Sequência ortogonal
	Time-Division Multiple Access (TDMA)

13.5 RESUMO

- Os métodos de acesso ao meio podem ser classificados como aleatório, ordenado ou canalizado.
- No método CSMA, uma estação deve primeiramente ouvir o meio para depois transmitir dados através dele.
- Uma estratégia de persistência define os procedimentos a serem seguidos quando uma estação verifica que o meio está ocupado.
- CSMA/CD é um método CSMA dotado de procedimentos pós-colisão.
- CSMA/CA é um método CSMA dotado de procedimentos para evitar colisão.
- Reserva, *polling* e passagem de permissão são métodos de acesso ordenado.
- No método de acesso com reserva, uma estação reserva um *slot* para dados assinalando o respectivo *flag* identificador no *frame* de reserva.
- No método de acesso *polling*, uma estação principal controla as transmissões para e das estações secundárias.
- No método de acesso por passagem de permissão, somente a estação que detém o controle do *frame*, denominado permissão, pode transmitir dados.
- Canalização é um método de acesso múltiplo no qual a banda disponível de um meio é compartilhada no tempo, frequência ou através de código entre as diferentes estações de uma rede.
- FDMA, TDMA e CDMA são métodos de canalização.
- No FDMA, a banda disponível é dividida em faixas, cada qual sendo reservada para o uso de uma estação específica.
- No TDMA, a banda disponível não é dividida em faixas. Em vez disso, toda a banda é compartilhada no tempo.
- No CDMA, a banda disponível não é dividida em faixas e, mesmo assim, os dados de todas as entradas são transmitidos simultaneamente.
- CDMA está baseada na teoria criptografia e utiliza sequências de números denominados *chips*. As sequências são geradas através da matriz de Walsh.

13.6 PRATIQUE OS CONHECIMENTOS ADQUIRIDOS

Questões de Revisão

1. Qual é a vantagem do acesso ordenado sobre o acesso aleatório?
2. Liste na sequência os protocolos que evoluíram do MA.
3. Quando devemos incrementar o *backoff* numa rede que utiliza o método ALOHA?
4. Como as duas estratégias de persistência diferem entre si?
5. Qual é o propósito do sinal de congestionamento (*jam signal*) no CSMA/CD?
6. De que maneira o método CSMA/CD difere do método CSMA/CA?

7. Cite os três métodos mais conhecidos de acesso ordenado.
8. O método de acesso com reserva é adequado a uma rede grande onde muitas estações podem se encontrar inativas (*idle*)? Por quê?
9. Cite a diferença entre *polling* e *selecting*.
10. Por que o método de passagem de permissão é um procedimento de acesso ordenado?
11. Cite os três protocolos de canalização.
12. Cite o número de faixas de frequência por largura de banda no FDMA, TDMA e CDMA.
13. De que maneira o método CDMA é superior ao FDMA? Como o CDMA é superior ao TDMA?
14. O que é uma colisão?
15. O que é o produto interno?

Questões de Múltipla Escolha

16. O primeiro método de acesso de que temos notícia é o _____.
 - ALOHA
 - CSMA
 - Canalização
 - Passagem de permissão
17. Não há colisão no método de acesso aleatório _____.
 - ALOHA
 - CSMA/CD
 - CSMA/CA
 - Passagem de permissão
18. As estações não ouvem o meio no método de acesso aleatório _____.
 - ALOHA
 - CSMA/CD
 - CSMA/CA
 - Ethernet
19. Na abordagem 1-persistent, quando uma estação encontra o meio livre ela _____.
 - Espera 0,1s antes de iniciar uma transmissão
 - Espera 1s antes de iniciar uma transmissão
 - Espera um tempo igual a $1-p$ antes de iniciar uma transmissão
 - Transmite imediatamente
20. Na abordagem p -persistent, quando uma estação encontra o meio livre ela _____.
 - Espera 1s antes de iniciar uma transmissão
 - Transmite com probabilidade $1-p$
 - Transmite com probabilidade p
 - Transmite imediatamente
21. Uma rede usando o método de acesso CSMA com $p = 0,25$ transmitirá _____ % do tempo após acessar um link livre.
 - 25
 - 50
22. A abordagem 1-persistent pode ser considerada um caso especial da abordagem p -persistent quando p é igual a _____.
 - 0,1
 - 0,5
 - 1,0
 - 2,0
23. _____ é um protocolo de acesso aleatório.
 - MA
 - Polling
 - FDMA
 - CDMA
24. _____ é um protocolo de acesso ordenado.
 - Acesso com reserva
 - FDMA
 - TDMA
 - CSMA
25. _____ é (são) protocolo(s) de canalização.
 - FDMA
 - TDMA
 - CDMA
 - Todos acima
26. _____ é o protocolo utilizado pelas redes Ethernet padrão.
 - CSMA
 - CSMA/CD
 - CSMA/CA
 - Passagem de permissão
27. Quando uma colisão é detectada numa rede usando CSMA/CD, _____.
 - O frame é retransmitido automaticamente
 - Um sinal de congestionamento (*jam signal*) é enviado pela estação transmissora

- c. O valor *backoff* é colocado em 0
d. O valor do *backoff* é decrementado de 1
28. Considerando o método de acesso com reserva, se existirem 10 estações numa rede, então são necessários _____ *minislots* de reserva no *frame* de reserva.
a. 5
b. 9
c. 10
d. 11
29. _____ requer uma estação principal e uma ou mais de uma estação secundária.
a. O acesso com reserva
b. O *polling*
c. A passagem de permissão
d. O CSMA
30. Quando o dispositivo principal indaga um dispositivo secundário se ele possui dados a transmitir, isto é denominado _____
a. *Polling*
b. *Selecting*
c. *Reserving*
d. *Backing off*
31. Se uma rede FDMA possui oito estações, a largura de banda do meio é dividida em _____ faixa(s).
a. 1
b. 2
c. 8
d. 16
32. Se um rede TDMA possui oito estações, a largura de banda do meio é dividida em _____ faixa(s).
a. 1
b. 2
c. 8
d. 16
33. Se um rede CDMA possui oito estações, a largura de banda do meio é dividida em _____ faixa(s).
a. 1
b. 2
c. 8
d. 16
34. A matriz de Walsh para 16 estações possui uma seqüência de _____ chips
a. 4
b. 8
c. 16
d. 32

Exercícios

35. Explique como o protocolo ALOHA responde a seguinte questão: quando a estação deve acessar o meio?
36. Explique como o protocolo ALOHA responde a seguinte questão: o que é feito se o meio estiver ocupado?
37. Explique como o protocolo ALOHA responde a seguinte questão: como a estação determina o sucesso ou falha da transmissão?
38. Explique como o protocolo ALOHA responde a seguinte questão: o que a estação faz se houver um conflito de acesso (colisão)?
39. Explique como o protocolo CSMA/CD responde a seguinte questão: quando a estação deve acessar o meio?
40. Explique como o protocolo CSMA/CD responde a seguinte questão: o que é feito se o meio estiver ocupado?
41. Explique como o protocolo CSMA/CD responde a seguinte questão: como a estação determina o sucesso ou falha da transmissão?
42. Explique como o protocolo CSMA/CD responde a seguinte questão: o que a estação faz se houver um conflito de acesso (colisão)?
43. Explique como o protocolo CSMA/CA responde a seguinte questão: quando a estação deve acessar o meio?
44. Explique como o protocolo CSMA/CA responde a seguinte questão: o que é feito se o meio estiver ocupado?
45. Explique como o protocolo CSMA/CA responde a seguinte questão: como a estação determina o sucesso ou falha da transmissão?
46. Explique como o protocolo CSMA/CA responde a seguinte questão: o que a estação faz se houver um conflito de acesso (colisão)?
47. Explique como o protocolo de passagem de permissão responde a seguinte questão: quando a estação deve acessar o meio?
48. Explique como o protocolo de passagem de permissão responde a seguinte questão: o que é feito se o meio estiver ocupado?
49. Explique como o protocolo de passagem de permissão responde a seguinte

- questão: como a estação determina o sucesso ou falha da transmissão?
50. Explique como o protocolo de passagem de permissão responde a seguinte questão: o que a estação faz se houver um conflito de acesso (colisão)?
 51. Complete a Tabela 13.1 para os diferentes protocolos discutidos neste capítulo. Responda sim ou não.
 52. Apresente a matriz de Walsh para W_{16} .
 53. Prove que a segunda propriedade das seqüências ortogonais para quaisquer duas entradas de sua escolha na W_{16} .
 54. Prove que a terceira propriedade das seqüências ortogonais para quaisquer duas entradas de sua escolha na W_{16} .
 55. Apresente a saída do multiplexador da Figura 13.16 se a estação 1 estiver em silêncio e todas as outras estações estiverem enviando um bit 1.
 56. Redesenhe a Figura 13.17 para o Problema 55.

TABELA 13.1 Exercício 51

<i>Característica</i>	<i>ALOHA</i>	<i>CSMA/CD</i>	<i>CSMA/CA</i>	<i>Passagem de permissão</i>	<i>Canalização</i>
Acesso Múltiplo					
<i>Carrier Sense</i>					
Detecta colisão					
Confirmação (ACK)					

Redes Locais Ethernet

No Capítulo 1 vimos que uma rede local (LAN) é uma rede de computadores desenvolvida para cobrir regiões geograficamente pequenas, tal como um prédio ou um campus. Hoje em dia, a maioria das LANs estão conectadas entre si formando *internetworks*, redes de longas distâncias (WANs) ou a Internet, embora uma LAN ainda possa ser utilizada como uma rede isolada conectando os computadores de uma empresa para o compartilhamento de recursos.

O mercado de LANs possui muitas tecnologias, mas a tecnologia predominante atualmente é a **Ethernet**. Neste capítulo focaremos as redes Ethernet. No Capítulo 15 examinaremos as Wireless LANs (WLANS).

A Figura 14.1 traz uma síntese comparativa de três gerações Ethernet.

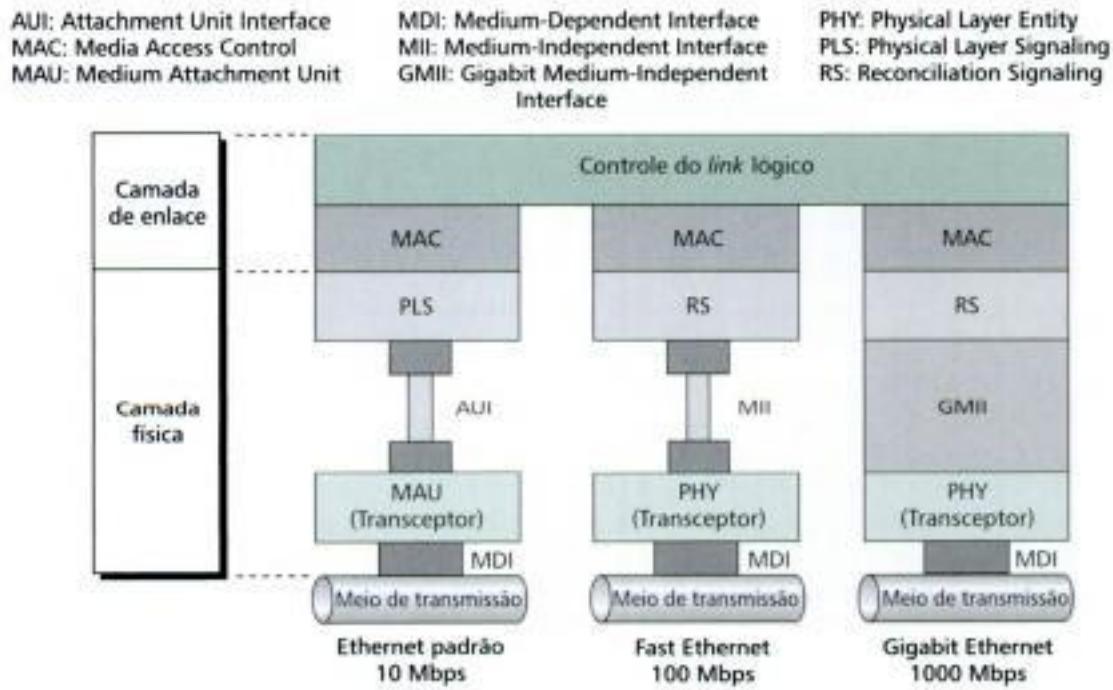


Figura 14.1 Três das gerações Ethernet.

A tecnologia Ethernet nasceu em 1976 no centro de pesquisas da Xerox (Palo Alto Research Center – PARC). Ela evoluiu muito desde então. Designaremos a Ethernet original, cuja taxa de transmissão é 10 Mbps, de Ethernet padrão. Os padrões mais atuais* Fast Ethernet e Gigabit Ethernet operam a 100 Mbps e 1 Gbps, respectivamente.

Um computador conectado à Internet via LAN precisa utilizar a arquitetura de cinco camadas da Internet. As três camadas mais altas (rede, transporte e aplicação) são comuns a todas as LANs. A camada de enlace de dados é dividida nas subcamadas de controle do link lógico (*Logical Link Control* – LLC) e de controle de acesso ao meio (*Medium Access Control* – MAC). A subcamada LLC foi desenvolvida originalmente para garantir a interoperabilidade de todas as LANs, mas não é usada com frequência atualmente. Em vez disso, a interoperabilidade é assegurada por um protocolo da camada de rede muito difundido, como veremos num capítulo adiante. Isto significa que as LANs diferem somente nas subcamadas MAC e nas camadas físicas. Enquanto a subcamada MAC é diferente apenas ligeiramente entre as tecnologias, a camada física é completamente diferente entre as versões da Ethernet.

14.1 ETHERNET PADRÃO

A Ethernet padrão foi desenvolvida para funcionar a 10 Mbps. O método de acesso ao meio adotado nas redes Ethernet padrão foi o CSMA/CD. Os meios são compartilhados entre todas as estações.

Subcamada MAC

A subcamada MAC governa toda a operação do método de acesso. Ela também recebe *frames* de dados da camada superior e os passa à subcamada PLS para codificação.

Método de Acesso: CSMA/CD

A Ethernet padrão usa CSMA/CD como método de acesso ao meio e estratégia de persistência 1-persistent. Os métodos de acesso foram analisados no Capítulo 13.

Frame

O *frame* Ethernet padrão possui sete campos: preâmbulo, SFD, DA, AS, comprimento/tipo de protocolo (PDU), dados da camada superior e CRC. A Ethernet não fornece nenhum mecanismo de confirmação dos *frames* recebidos. As confirmações (ACKs) devem ser implementadas nas camadas mais altas. A Figura 14.2 mostra um *frame* MAC da rede Ethernet padrão.

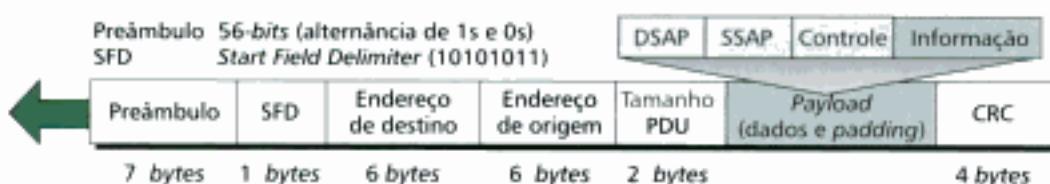


Figura 14.2 Frame MAC 802.3.

- **Preâmbulo.** O primeiro campo do *frame* 802.3 possui 7 bytes (56 bits) formando um padrão alternado de 1s e 0s para sincronização do sistema receptor. O padrão proporciona somente um alerta e pulso de relógio (*timing*). O padrão de 56 bits permite que as estações percam alguns bytes de início e mesmo assim ajustem os relógios internos de sincronismo. De fato, o **preâmbulo** é adicionado ao *frame* na camada física e, formalmente, não faz parte do *frame*.

* N. de R. T.: Até o presente instante, existe em funcionamento, além da Fast e Gigabit Ethernet supracitados, a 10 Gigabit Ethernet. A Cisco está desenvolvendo ou sugerindo um novo padrão, denominado Metro Ethernet, para operar inicialmente a 40 Gbps e interligar LANs e WANs.

- **Campo delimitador de início de quadro (Start Frame Delimiter – SFD).** O segundo campo de 1-byte de comprimento (byte: 10101011) sinaliza o início do frame. O SFD informa às estações que elas têm uma última chance de obter sincronização. Os dois últimos bits são 11 e alertam o receptor que o próximo campo contém o endereço de destino do frame.
- **Endereço de destino (Destination Address – DA).** O campo DA tem 6-bytes e contém o endereço físico da estação de destino ou a estação que deve receber o pacote. Analisaremos o campo de endereço de destino mais detalhadamente.
- **Endereço de origem (Source Address – SA).** O campo SA também possui 6-bytes de comprimento e contém o endereço físico da estação que originou a transmissão do pacote. Analisaremos o campo de endereço de origem mais detalhadamente.
- **Comprimento/tipo de protocolo.** Este campo é definido como campo de comprimento ou tipo de protocolo. Se o valor do campo é menor que 1518, ele é um campo de comprimento que define o tamanho do campo de dados (próximo campo). De outro modo, se o valor desse campo é superior a 1536, ele define o tipo de pacote PDU (Protocol Data Unit) encapsulado no frame.
- **Dados (payload).** Este campo transporta os dados encapsulados pelos protocolos das camadas superiores. Como veremos, ele possui um mínimo de 46 e um máximo de 1500 bytes.
- **CRC.** O último campo carrega informação sobre detecção de erro, neste caso um CRC-32.

Tamanho do Frame O padrão Ethernet impõe restrições tanto ao tamanho mínimo quanto o máximo de um frame, conforme ilustra a Figura 14.3

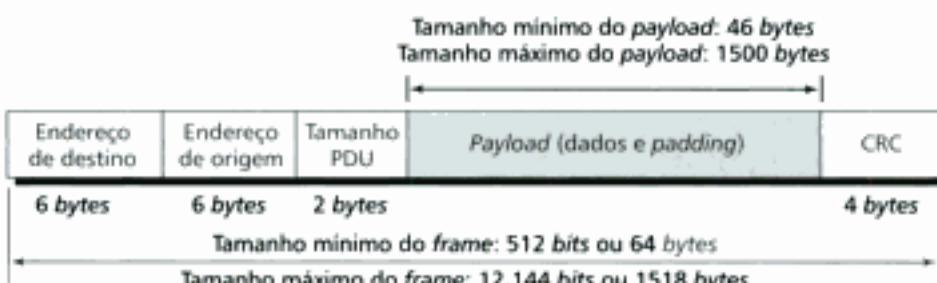


Figura 14.3 Tamanho mínimo e máximo.

A restrição de tamanho mínimo é requerida para o perfeito funcionamento do método CS-MAC/CD. Se ocorrer uma colisão antes da camada física enviar um frame para fora da estação, ela deve ser ouvida por todas as estações. Se todo o frame for enviado antes da colisão ser detectada é tarde demais e nada há de ser feito. A subcamada MAC descarta de fato o frame inferindo que o frame alcançou o destino. Esta situação é agravada quando o tamanho do frame é reduzido em direção ao tamanho mínimo. Sendo assim, o menor frame padronizado para LANs Ethernet de 10 Mbps é 512-bits ou 64-bytes (sem o campo préambulo ou SFD).

Um frame Ethernet deve possuir um tamanho mínimo de 512-bits ou 64-bytes. Parte disso é cabeçalho e trailer. Se contarmos 18-bytes de cabeçalho e trailer (6 bytes endereço de origem, 6 bytes endereço de destino, 2 bytes tamanho/tipo de protocolo e 4 bytes CRC), então o tamanho mínimo do campo de dados da camada superior (payload) será $64 - 18 = 46$ bytes. Se os dados da camada superior forem menores 46 bytes são adicionados bytes de enchimento (padding) para completar o total. O padrão define o tamanho máximo do frame (sem os campos de préambulo e SFD) como 1518 bytes. O tamanho máximo do payload é 1500 bytes se subtraímos os 18 bytes de cabeçalho e trailer. Esse valor para a restrição no tamanho máximo do frame tem origens históricas.

Endereçamento

Cada estação numa rede Ethernet (tal como um PC, *workstation* ou impressora) possui seu próprio **adaptador de rede** ou **Network Interface Card (NIC)**. O NIC é encaixado dentro da estação e fornece à estação um endereço físico de 6-bytes. Logo, o endereço Ethernet tem 6-bytes (48-bits) que são escritos normalmente em **notação hexadecimal** separada por hífens entre os bytes, conforme ilustra a Figura 14.4.

06-01-02-01-2C-4B

Figura 14.4 Endereço físico (MAC) notação hexadecimal.

Endereços Unicast, Multicast e Broadcast O endereço de origem é sempre **unicast**, ou seja, o frame origina numa única estação. Entretanto, o endereço de destino pode ser **unicast**, **multicast** ou **broadcast**. A Figura 14.5 mostra como distinguir um endereço *unicast* de um endereço *multicast*.



Figura 14.5 Endereços *unicast* e *multicast*.

Um endereço de destino *unicast* define somente um receptor, isto é, o relacionamento entre transmissor e receptor é único (dedicado). Um endereço de destino *multicast* define um grupo de endereços receptores, isto é, o relacionamento entre transmissor e receptor é múltiplo. O endereço de *broadcast* é um caso especial de endereço *multicast*. Os receptores são todas as estações de uma rede (público). Um endereço de *broadcast* tem 1s em todos os 48-bits do campo.

Camada Física

A Figura 14.6 ilustra o modelo da camada física da Ethernet de 10 Mbps.

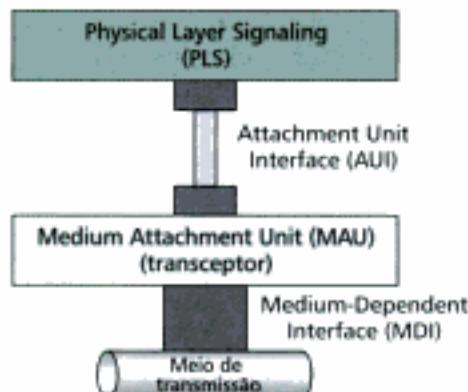


Figura 14.6 Camada física.

PLS

A **subcamada PLS (Physical Layer Signaling)** codifica e decodifica dados. A Ethernet padrão usa o método de codificação Manchester (veja Capítulo 4) a uma taxa de 10 Mbps. Note que é necessária uma taxa de modulação de 20 Mbaud. A Figura 14.7 mostra as funcionalidades da subcamada PLS.

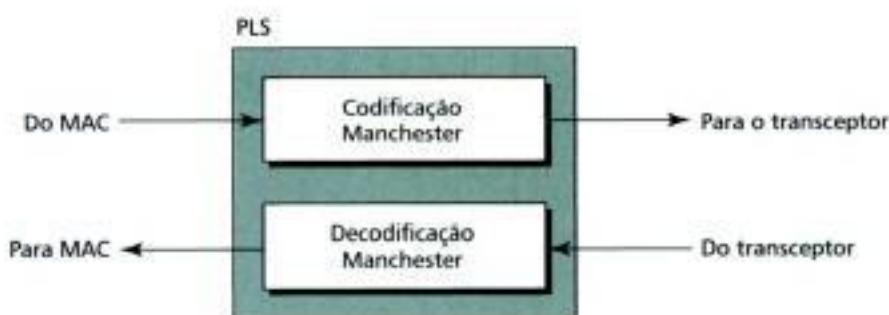


Figura 14.7 PLS.

AUI

A interface **AUI (Attachment Unit Interface)** é uma especificação que define a interface entre os níveis PLS e MAU. A interface AUI foi desenvolvida para criar um tipo de *interface independente do meio* entre PLS e MAU. Inicialmente, essa interface foi projetada para a primeira versão da Ethernet padrão que utilizava cabo coaxial. A idéia central era padronizar a conexão da subcamada PLS a MAU de modo que, se no futuro, fosse necessário usar uma MAU diferente (usando outro meio de transmissão), não era necessário trocar a PLS. A Figura 14.8 é uma ilustração da interface, cabo e conector AUI.

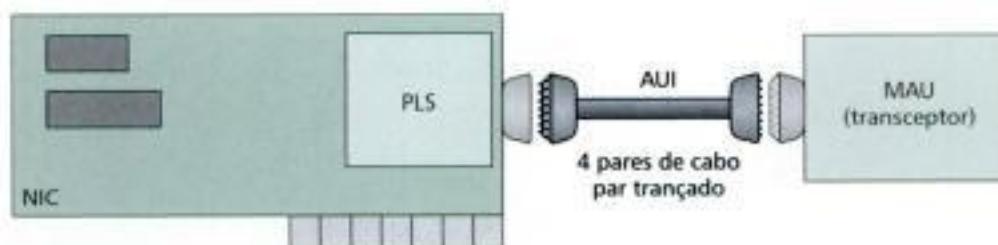


Figura 14.8 AUI

MAU (Transceptor ou Transceiver)

O **Medium Attachment Unit (MAU)** é dependente do meio. O MAU produz um sinal apropriado para cada meio em particular. Há um MAU para cada tipo meio usado na Ethernet de 10 Mbps. O cabo coaxial tem um MAU particular, o cabo par trançado e a fibra óptica têm outras.

O **transceptor** é um dispositivo que condensa o transmissor e o receptor. Ele possui a capacidade de transmitir sinais no meio e receber sinais do meio. Outra funcionalidade do transceptor é detectar colisões. A Figura 14.9 mostra a posição e as funcionalidades do transceptor.

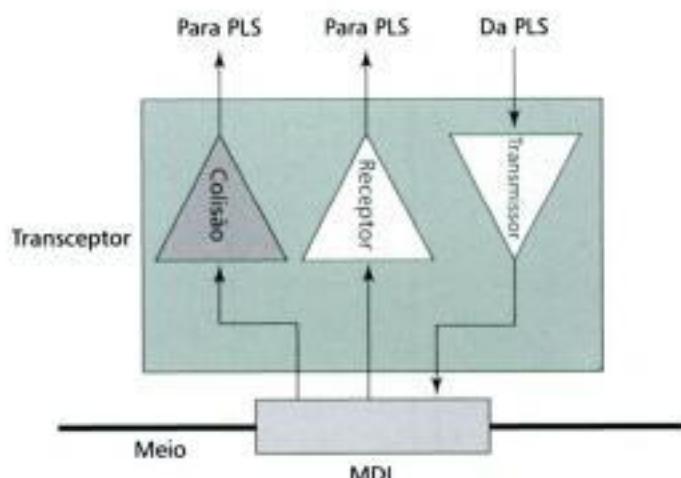


Figura 14.9 MAU (transceptor).

Um transceptor pode ser um dispositivo interno ou externo. O transceptor externo é instalado junto ao meio e é conectado à estação via uma interface AUI. Por outro lado, o transceptor interno reside dentro da estação (na própria interface) e não necessita de cabo AUI.

MDI

Para conectar o transceptor, interno ou externo, ao meio necessitamos de uma interface **MDI (Medium-Dependent Interface)**. A MDI é apenas parte do *hardware* de conexão do transceptor ao meio. Para a instalação do transceptor externo pode ser utilizado um conector tipo T ou *tap*. Para um transceptor interno pode ser utilizado um *jack*.

Implementação da Camada Física

A Ethernet padrão (10 Mbps) define quatro diferentes tipos de implementação em banda base (digital), isto é, não utiliza modulação, conforme ilustra a Figura 14.10. Examinaremos cada tipo de implementação separadamente.



Figura 14.10 Implementações Ethernet padrão.

10Base5: Thick Ethernet

A primeira implementação foi batizada de **10Base5, thick Ethernet** ou **Thicknet**. O apelido deriva do tipo de cabo utilizado no sistema de cabeamento, o qual é grosseiramente do tamanho de uma típica mangueira utilizada na jardinagem e muito rígido para se curvar utilizando-se as mãos. O padrão 10Base5 foi a primeira especificação Ethernet.

A topologia utilizada em redes padrão 10Base5 é a barramento ou linear que utiliza transceptor externo conectado via *tap* ao cabo coaxial grosso. A Figura 14.11 ilustra o modo de conexão de uma estação ao meio.

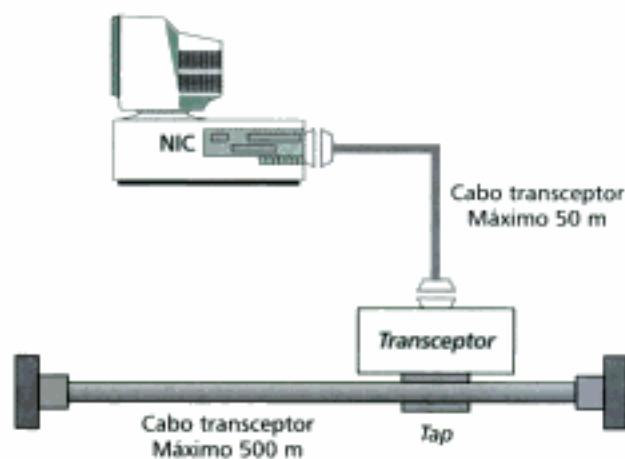


Figura 14.11 Conexão de uma estação ao meio usando padrão 10Base5.

10Base2: Thin Ethernet

A segunda implementação é denominada **10Base2, thin Ethernet**, ou *Cheapnet*. O padrão 10Base2 também utiliza topologia em barramento, incorpora transceptor internamente ou um emprega

um transceptor externo, via conexão ponto a ponto. A Figura 14.12 ilustra a conexão de duas estações ao meio. Perceba que se a estação utilizar um transceptor interno, não há necessidade do cabo AUI. Se a estação carece do transceptor, então pode ser utilizado um transceptor externo juntamente com o *kit AUI*.

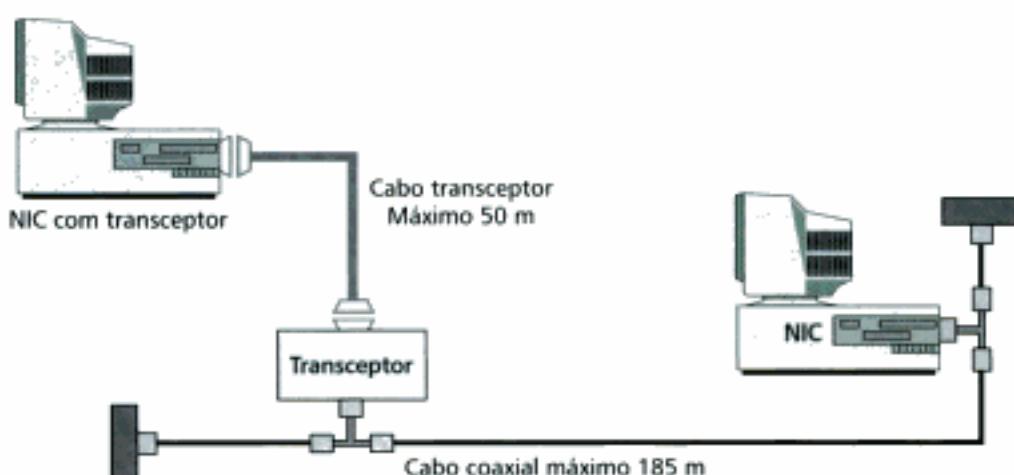


Figura 14.12 Conexão das estações ao meio usando padrão 10Base2.

10BaseT: Ethernet par trançado

A terceira implementação é denominada **10BaseT** ou **Ethernet par trançado**. O padrão 10BaseT utiliza topologia física em estrela. As estações são conectadas a um *hub* que pode utilizar transceptor interno ou externo. Quando o transceptor interno é utilizado não há necessidade do cabo AUI, a interface de rede é conectada diretamente ao meio através de conector (tipicamente o RJ45). O transceptor é então conectado ao concentrador, conforme ilustra a Figura 14.13.

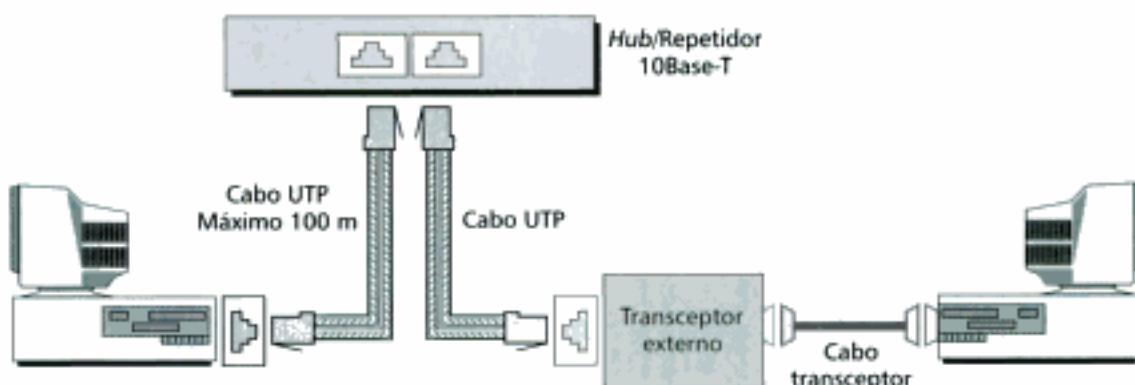


Figura 14.13 Conexão das estações ao meio usando padrão 10Base-T.

10BaseFL: Fiber Link Ethernet

Embora existam muitos tipos de implementação Ethernet utilizando fibra óptica para transmissão em 10 Mbps, um excelente padrão definido pelos fabricantes é denominado **10BaseFL** ou **Fiber Link Ethernet**. O padrão 10BaseFL utiliza topologia estrela para conectar as estações a um *hub*. O padrão é implementado normalmente através de transceptor externo, denominado MAU óptico. Assim, a estação é conectada ao transceptor externo via cabo AUI. Por sua vez, o transceptor é conectado ao *hub* através de dois pares de cabos de fibra óptica, conforme ilustra a Figura 14.14.

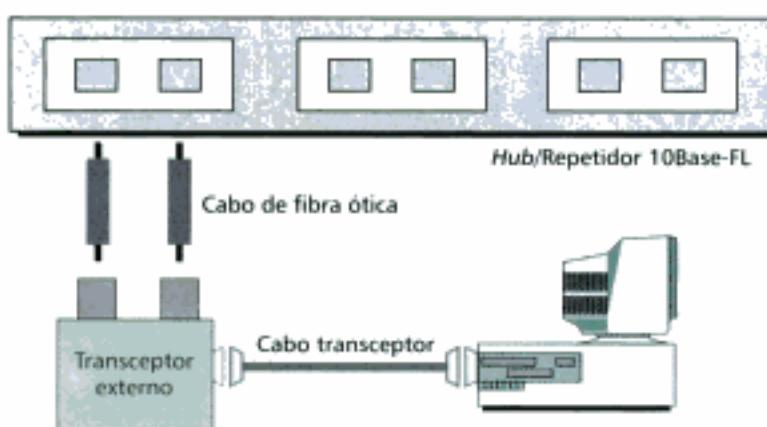


Figura 14.14 Conexão das estações ao meio usando padrão 10Base-FL.

Ethernet Interligada por Bridges

A primeira etapa da evolução das redes Ethernet foi a divisão das LANs por *bridges*. As *bridges* produzem dois efeitos sobre as LANs Ethernet: elas aumentam a banda do segmento de rede e separam domínios de colisão. No Capítulo 16 discutiremos detalhes sobre o funcionamento das *bridges*.

Aumentando a Banda

Numa rede Ethernet sem *bridge* a capacidade total (10 Mbps) é compartilhada entre todas as estações que têm *frame*(s) a transmitir, isto é, as estações compartilham a largura de banda da rede. Se uma única estação tiver *frames* a transmitir ela se beneficia da capacidade total do meio (10 Mbps). Mas, como foi dito, se houver mais de uma estação tentando transmitir na rede a capacidade deve mesmo ser compartilhada. Por exemplo, se duas estações tiverem muitos *frames* a transmitir elas provavelmente alternam o uso do meio. Quando uma estação tiver transmitindo, a outra estação está aguardando a vez dela. Podemos dizer que, na média, cada estação transmite numa taxa de 5 Mbps. A Figura 14.15 ilustra a situação.

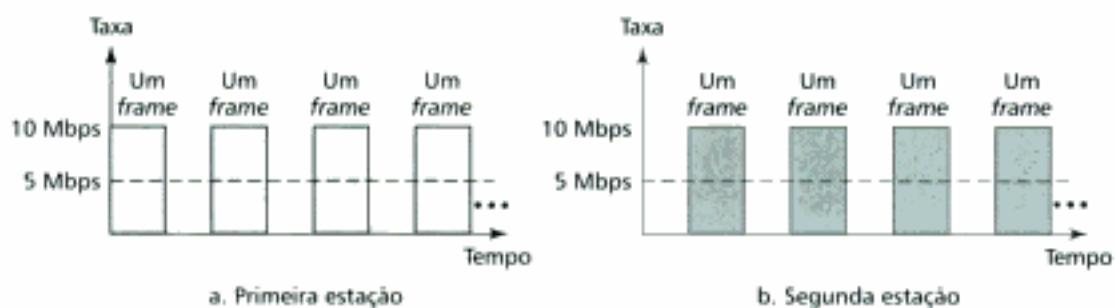


Figura 14.15 Compartilhamento da banda.

A *bridge*, conforme veremos no Capítulo 16, possui um papel importante nestas circunstâncias. Uma *bridge* divide uma rede original em dois ou mais segmentos de rede. Olhando a banda de cada rede, elas são totalmente independentes. Por exemplo, na Figura 14.16, uma rede com 12 estações é dividida em duas redes menores, cada qual com 6 estações. Cada uma dessas redes tem uma capacidade máxima de 10 Mbps. Em vez de compartilhar essa capacidade (10 Mbps) entre 12 estações, ela a compartilha entre 6 estações em cada rede (a rigor são 7 porque uma *bridge* funciona como uma estação em cada segmento de rede). Numa rede com tráfego de dados intenso, cada estação tem, teoricamente, 10/6 Mbps de banda ao invés de 10/12 Mbps, assumindo que o tráfego não acontece através da *bridge*.

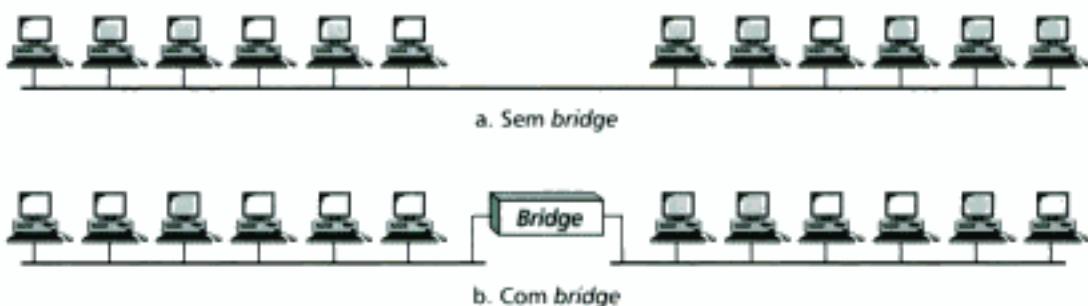


Figura 14.16 Um rede com e sem bridge.

Claro que, se continuarmos a dividir a rede, ganharemos mais banda em cada segmento. Por exemplo, se utilizarmos uma *bridge* de quatro portas, cada estação terá 10/3 Mbps de banda para transmissão, o que significa 4 vezes mais que a rede inicial sem a *bridge*.

Separando Domínios de Colisão

Outra vantagem de uma *bridge* é a separação ou quebra do **domínio de colisão**. A Figura 14.17 mostra os domínios de colisão para duas redes: uma sem, outra com *bridge*. Você pode notar que o domínio de colisão foi quebrado e a probabilidade de colisão reduzida tremendoamente. Sem *bridge*, temos um total de 12 estações acessando o meio. Com *bridge*, somente 3 estações acessam o segmento de rede a qual estão conectadas.

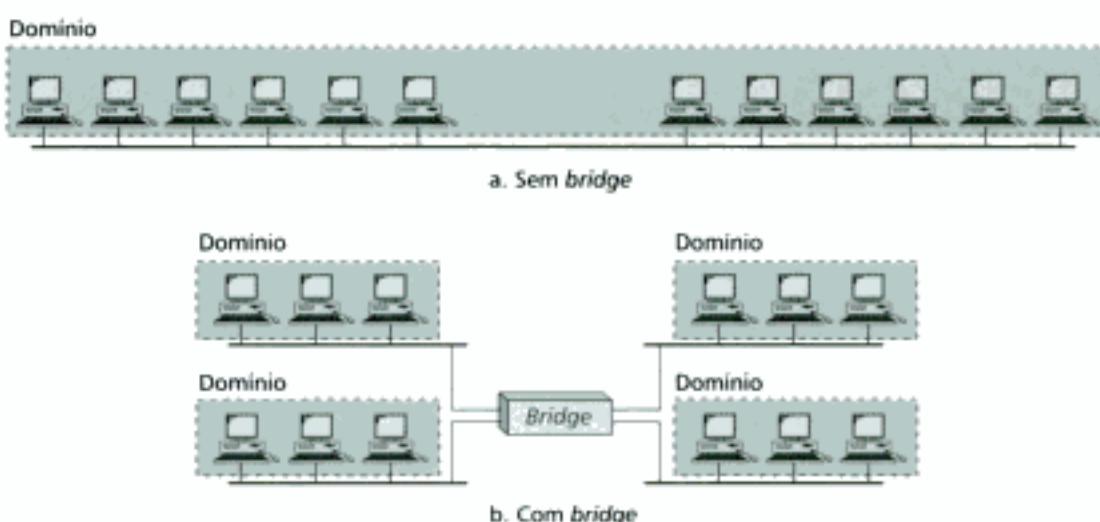


Figura 14.17 Domínios de colisão numa rede com e sem bridge.

Switched Ethernet

A idéia de uma LAN interligada por *bridges* pode ser estendida à LAN interligada por *switches*. Em vez de dois ou quatro segmentos de rede, porque não N segmentos, onde N é o número de estações na LAN? Noutras palavras, se dispusermos de uma *bridge* multiportas poderíamos levar isso adiante. Entretanto, como veremos, melhor do que uma *bridge* multiportas é utilizar um *switch* de N portas. Desse modo, a banda é compartilhada somente entre a estação e o *switch* (5 Mbps para cada). Adicionalmente, o domínio de colisão original é quebrado em N domínios menores.

Um **switch** de camada 2 é uma *bridge* com N portas com algumas sofisticações adicionais que permitem um tratamento melhor dos pacotes. Como veremos, a evolução de uma Ethernet interligada por *bridges* para a **Switched Ethernet** foi um enorme passo em direção aos padrões Ethernet mais velozes. A Figura 14.18 ilustra uma LAN controlada por *switch*.

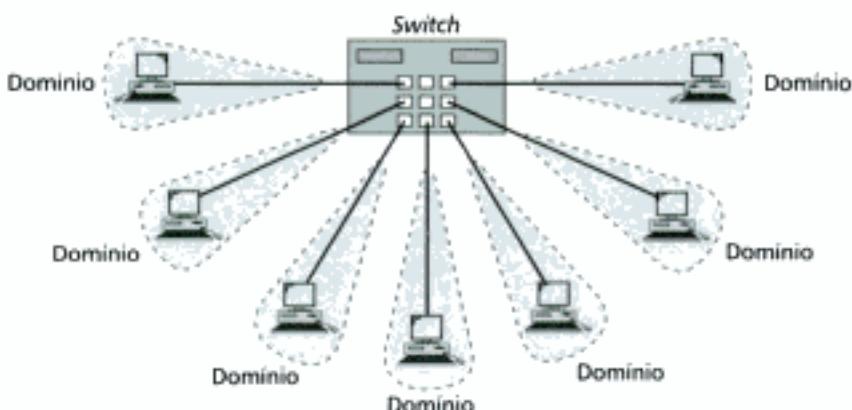


Figura 14.18 Switched Ethernet.

Ethernet Full-Duplex

Uma das limitações do padrão 10Base5 e 10Base2 é que a comunicação acontece no modo *half-duplex* (10BaseT é sempre *full-duplex*). Assim, uma estação pode transmitir ou receber, mas nunca os dois ao mesmo tempo. O próximo passo na evolução foi migrar das redes Ethernet com *switch* para as redes **full-duplex Switched Ethernet**. O modo *full-duplex* melhora a capacidade de cada segmento de 10 para 20 Mbps. A Figura 14.19 ilustra uma rede Ethernet com *switch* em modo *full-duplex*. Note que, ao invés de utilizar um único link entre uma estação e o *switch*, a configuração utiliza dois links, um para transmitir e o outro para receber.

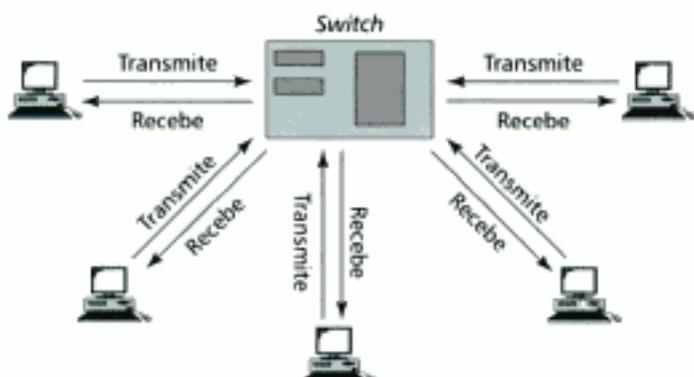


Figura 14.19 Rede Switched Ethernet full-duplex.

Acesso sem CSMA/CD

Na Ethernet com *switch* em modo *full-duplex* não há necessidade do método CSMA/CD. Nele, as estações são conectadas ao *switch* via dois *links* separados. Sendo assim, cada estação pode transmitir e receber independentemente, sem se preocupar com as colisões. De outro modo, cada *link* é um acesso dedicado (ponto a ponto) entre a estação e o *switch*. Não há mais necessidade da portadora "ouvir" o cabo ou necessidade de detecção de colisão. O trabalho da subcamada MAC torna-se ainda mais fácil. As funcionalidades "ouvir" o cabo e detectar colisão podem ser desligadas na subcamada MAC.

MAC

A Ethernet padrão foi desenvolvida com um protocolo sem conexão à subcamada MAC. Não existe nenhum controle de fluxo ou controle de erros explícito para informar ao transmissor que o *frame* chegou ao destino livre de erro. Quando o receptor captura o *frame* não envia nenhum tipo de confirmação, positivo (ACK) ou negativo (NACK).

Para agregar mecanismos de controle de fluxo e erro na Ethernet com *switch* em modo *full-duplex*, uma nova camada, denominada MAC controle, foi adicionada entre as subcamadas LLC e MAC.

14.2 FAST ETHERNET

A Ethernet padrão foi uma revolução nas LANs, mas é claro que, à medida que as aplicações de rede iam sendo desenvolvidas, novas demandas por melhores taxas de transmissão foram surgindo. Assim surgiu o protocolo **Fast Ethernet** (100 Mbps).

Subcamada MAC

Toda idéia por detrás da evolução da Ethernet de 10 para 100 Mbps foi manter intacta a subcamada MAC. O método de acesso é o mesmo (CSMA/CD). Além disso, é claro que para as redes Fast Ethernet com *switch* em modo *full-duplex* não é necessário o CSMA/CD. Entretanto, as implementações mantiveram CSMA/CD para assegurar a compatibilidade com a Ethernet padrão. O formato do *frame*, tamanhos mínimo e máximo e o endereçamento são os mesmos para Ethernet de 10 e 100 Mbps.

Autonegotiação

Uma nova característica agregada ao padrão Fast Ethernet foi a **autonegotiação**. Ela permite a uma estação ou *hub* uma faixa de capacidade de transmissão. A autonegotiação possibilita a dois dispositivos negociar o modo ou a taxa de transmissão de dados. Ela foi desenvolvida particularmente para os seguintes propósitos:

- Permitir a conexão entre dispositivos incompatíveis. Por exemplo, um dispositivo com capacidade máxima de 10 Mbps consegue se comunicar com outro dispositivo projetado para 100 Mbps, capaz de trabalhar em velocidades menores.
- Possibilitar várias velocidades de transmissão a um dispositivo.
- Permitir a uma estação sondar a capacidade do *hub*.

Camada Física

A Figura 14.20 mostra o esquema de camadas para a Ethernet 100 Mbps.

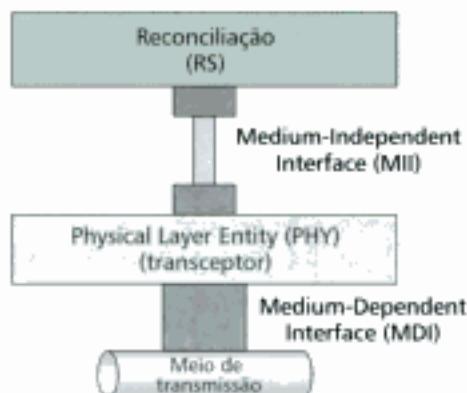


Figura 14.20 Camada física Fast Ethernet.

A camada física foi montada a partir de quatro subcamadas, a saber: RS, MII, PHY e MDI. A subcamada de reconciliação (RS) é comum a todas implementações. As subcamadas PHY e MDI dependem do meio de transmissão.

RS

No padrão Fast Ethernet, a **subcamada de reconciliação** substitui a subcamada PLS da Ethernet padrão. As funções de codificação e decodificação, as quais eram realizadas na PLS, foram movidas para a subcamada PHY (transceptor) visto que a codificação Fast Ethernet é dependente do meio. A subcamada RS é responsável por tudo que é deixado pelas demais subcamadas, especificamente, a passagem de dados no formato de *nibble* (4-bits) para a MII, como veremos em breve.

MII

Durante o desenvolvimento da Fast Ethernet, a interface AUI foi substituída pela **MII (Medium-Independent Interface)**. A MII é uma interface bastante melhorada que pode ser utilizada tanto em taxas de 10 quanto 100 Mbps. A Figura 14.21 ilustra a MII.

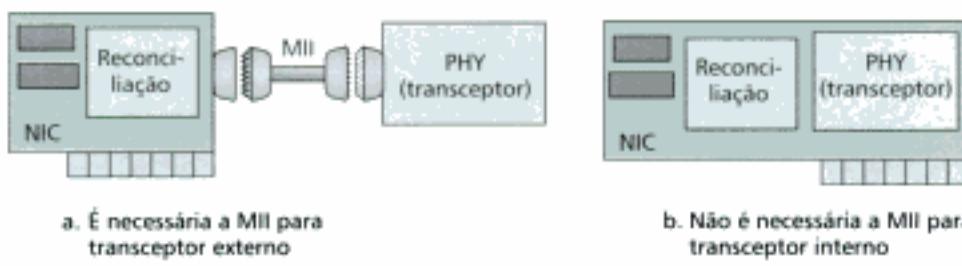


Figura 14.21 MII.

Abaixo, sintetizamos as características da MII:

- Opera tanto em 10 quanto 100 Mbps. Noutras palavras, ela é totalmente compatível com a interface AUI.
- Proporciona a conexão paralela de dados (*4-bits* por vez) entre as subcamadas PHY e RS.
- Agrega funções de gerenciamento.

PHY (Transceptor)

O transceptor da rede Fast Ethernet recebe o nome de **subcamada PHY**. Além das funções regulares mencionadas na rede Ethernet padrão, o transceptor é responsável pela codificação e decodificação. Esta função foi deslocada da subcamada PLS para a PHY. Novamente, o transceptor pode ser interno ou externo. Um transceptor externo é instalado junto ao meio, sendo conectado a uma estação via cabo MII. Um transceptor interno reside na estação, na interface de rede, e não é necessário o cabo MII. Os usuários têm predileção em adquirir interfaces que incorporem todos os recursos para simplificar as coisas. Assim, o transceptor interno é o predileto nas redes Fast Ethernet. Além do que, como um transceptor é sensível ao meio, deixaremos essa discussão para a próxima seção, que tratará as implementações.

MDI

Para conectar um transceptor (interno ou externo) ao meio necessitamos da Medium-Dependent Interface (MDI). A MDI é apenas parte do *hardware* que é específica da implementação.

Implementação da Camada Física

As implementações Fast Ethernet podem ser classificadas de acordo com a forma de conexão: dois ou quatro pares. A implementação a dois pares é denominada 100Base-X, a qual pode ser par trançado (100Base-Tx) ou fibra óptica (100Base-FX). A implementação a quatro pares foi desenvolvida apenas para par trançado (100Base-T4). Noutras palavras, temos três implementações: 100BaseTX, 100Base-FX e 100Base-T4, conforme ilustra a Figura 14.22.

100Base-TX

O padrão **100Base-TX** usa dois pares do cabo par trançado (UTP ou STP categoria 5) na topologia estrela. A implementação permite tanto transceptor interno quanto externo (utilizando o cabo MII). A Figura 14.23 mostra os dois tipos de conexão.

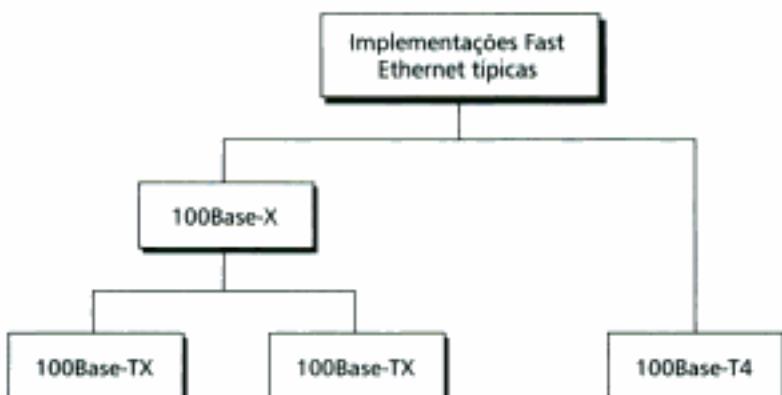


Figura 14.22 Implementações Fast Ethernet.

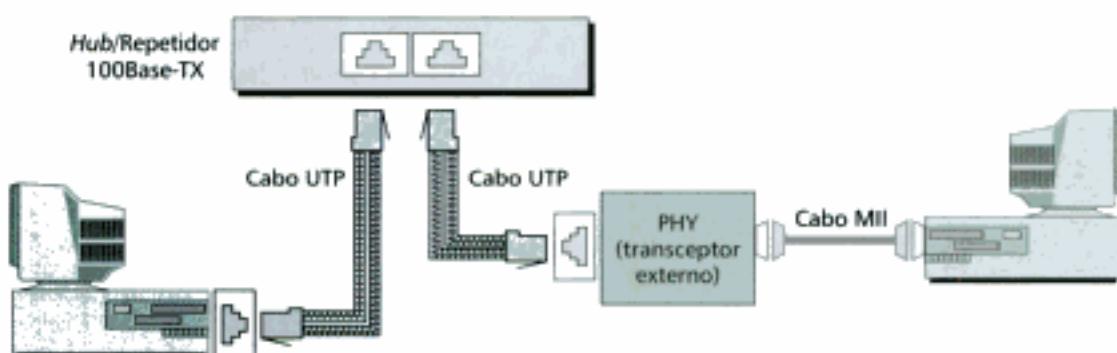


Figura 14.23 Implementação 100Base-TX.

Transceptor Nas redes Fast Ethernet, o transceptor é responsável pela transmissão, recepção, detecção de colisões e codificação/decodificação de dados.

Codificação e Decodificação Para realizar transmissão de dados a 100 Mbps, a codificação e decodificação é implementada em duas etapas, conforme Figura 14.24.

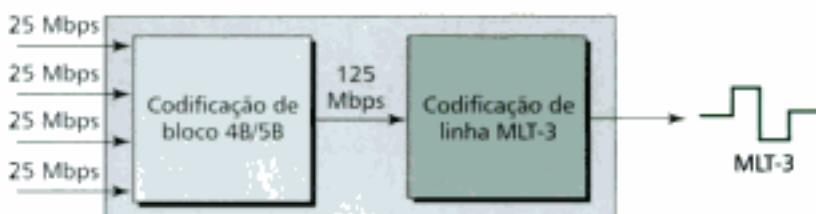


Figura 14.24 Codificação e decodificação no padrão 100Base-TX.

De modo a manter o sincronismo, o codificador realiza primeiramente a codificação dos dados. Os 4-bits que chegam paralelamente do NIC são codificados para serial usando a codificação 4B/5B discutida no Capítulo 4. Isto requer uma largura de banda de 125 MHz (125 Mbps).

Os dados a 125 Mbps são então codificados num sinal utilizando MLT-3 (veja Capítulo 4).

100Base-FX

O padrão **100Base-FX** usa dois pares de cabos de fibra óptica numa topologia física estrela. A implementação permite tanto o uso do transceptor interno quanto o externo (utilizando o cabo MII). A Figura 14.25 ilustra os dois tipos de implementação.

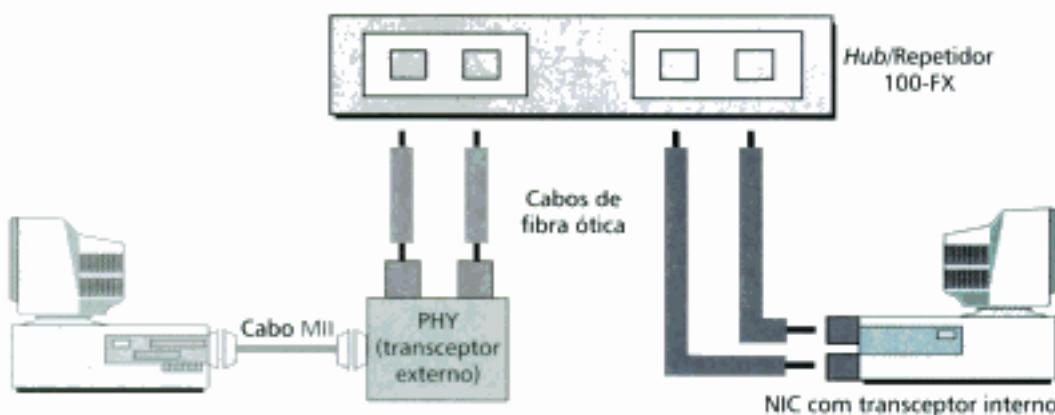


Figura 14.25 Implementação 100Base-FX.

Transceptor O transceptor é responsável pela transmissão, recepção, detecção de colisões e codificação/decodificação.

Codificação e Decodificação O padrão 100Base-FX utiliza dois níveis de codificação, conforme ilustra a Figura 14.26.

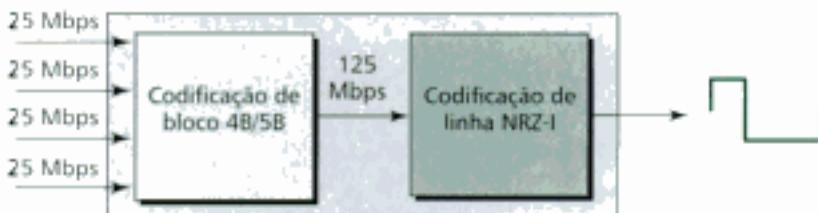


Figura 14.26 Codificação e decodificação no padrão 100Base-FX.

Para manter a sincronização, o codificador realiza primeiramente codificação de blocos. Os 4-bits que chegam paralelamente do NIC são codificados para serial usando a codificação 4B/5B discutida no Capítulo 4. Isto requer uma largura de banda de 125 MHz (125 Mbps).

Os dados a 125 Mbps são então codificados num sinal utilizando NRZ-I (veja Capítulo 4).

100Base-T4

O padrão 100Base-TX proporciona taxas de comunicação de 100 Mbps, mas requer o uso de cabos UTP ou STP categoria 5. Alguns anos atrás, boa parte do cabeamento das empresas baseava-se no cabo par trançado categoria 3 para canal de voz. Por isso, a substituição do cabeamento não era muito atraente em termos de custo/benefício na migração para o padrão 100Base-TX. O padrão **100Base-T4** foi desenvolvido para utilizar essa capacidade instalada, pois utiliza cabos categoria 3 ou superior. Essa implementação usa quatro pares UTP para transmissão a 100 Mbps. Entretanto, atualmente, a maior parte do cabeamento está baseada no cabo UTP CAT 5, o que praticamente garantiu a adoção do padrão 100Base-TX, em detrimento do 100Base-T4. A Figura 14.27 ilustra a conexão de uma estação numa rede 100Base-T4.

Transceptor

A função do transceptor no padrão 100Base-T4 é similar às outras implementações. Entretanto, a codificação/decodificação é muito mais complexa.

Codificação e Decodificação Para manter a sincronização e ao mesmo tempo reduzir a utilização de banda é adotada a codificação 8B/6T (veja Capítulo 4).

Transmissão a Quatro Fios A codificação 8B/6T reduz o uso da banda de 100 para 75 Mbaud (razão 8/6). O 100Base-T4 foi desenvolvido para operar em larguras de banda de 25 Mbaud. Em

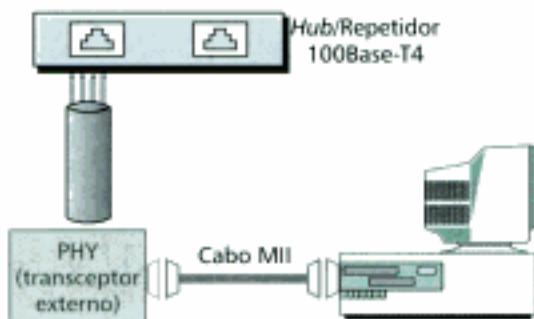


Figura 14.27 Implementação 100Base-T4.

transmissões unidirecionais isso iria requerer seis pares de cabo (três pares em cada direção). Para reduzir o número de pares para quatro, dois pares foram utilizados para transmissão unidirecional e os outros dois para transmissão bidirecional. Os dois pares unidirecionais estão sempre livres numa direção para transmitir sinais de colisão. A Figura 14.28 ilustra a sequência de fios.

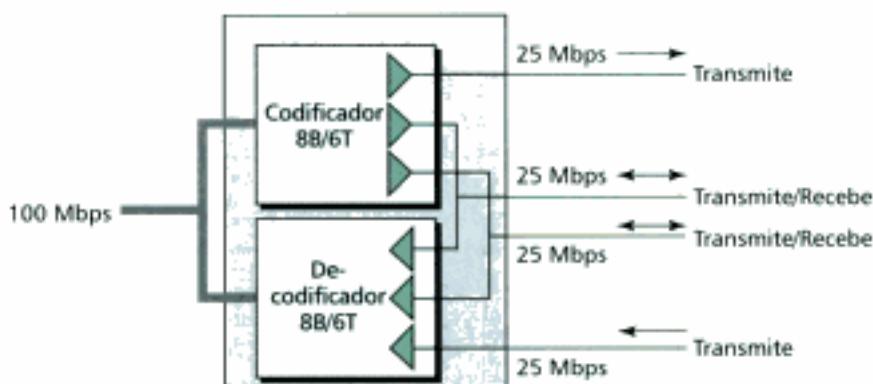


Figura 14.28 100Base-T4: padrão a quatro fios.

14.3 GIGABIT ETHERNET

Os avanços recentes de aplicações de alta resolução 3D, vídeo em tempo real, serviços de publicação (editoração digital) e outros resultaram no protocolo **Gigabit Ethernet** (1000 Mbps).

Subcamada MAC

Outra vez, a idéia principal por detrás da evolução da Ethernet era manter intacta a subcamada MAC. Entretanto, como veremos, decorriam alguns problemas quando as taxas de 1 Gbps eram solicitadas.

Método de Acesso

O padrão Gigabit Ethernet tem duas opções para o método de acesso ao meio: modo *half-duplex* com CSMA/CD ou modo *full-duplex* sem CSMA/CD. Embora a opção *half-duplex* seja muito interessante, ela se mostrou muito complicada e não é utilizada atualmente. Na opção *full-duplex* não há necessidade do método CSMA/CD. Quase todas implementações da Gigabit Ethernet seguem a opção em modo *full-duplex*.

Camada Física

A Figura 14.29 mostra a camada física da Gigabit Ethernet. Ela é dividida em quatro subcamadas: RS, GMII, PHY e MDI. A subcamada de reconciliação é comum a todas as implementações. As subcamadas PHY e MDI são dependentes do meio. Nessa seção, faremos uma breve discussão destas subcamadas. Na próxima seção, definiremos GMII, PHY e MDI para cada implementação particular.

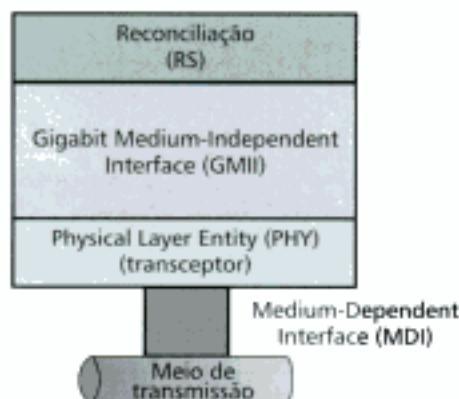


Figura 14.29 Camada física Gigabit Ethernet.

RS

A subcamada de reconciliação (RS) transmite, paralelamente, 8-bits de dados para a subcamada PHY via interface GMII.

GMII

GMII (Gigabit Medium-Independent Interface) é uma especificação que define como a subcamada RS é conectada à subcamada PHY (transceptor). Ela é a contrapartida da MII na Fast Ethernet. Entretanto, a GMII não tem a componente física externa, ou seja, ela não existe fora da NIC. Noutras palavras, a GMII constitui-se muito mais numa interface lógica do que física. Ela é uma especificação para os circuitos integrados (*chips*) da Gigabit Ethernet da interface NIC. Algumas características da GMII são:

- Ela opera somente a 1000 Mbps. Entretanto, existem *chips* que suportam tanto MII quanto GMII. Nesses casos, uma estação dispõe de autonegotiação e, assim, pode se conectar a 10, 100 e 1000 Mbps.
- A GMII especifica transmissão paralela de dados em 8-bits entre a subcamada RS e o transceptor.
- São agregadas funções de gerenciamento.
- Não existe cabo GMII.
- Não existe conector GMII.

PHY (Transceptor)

Exatamente como na Fast Ethernet, o transceptor é dependente do tipo de meio e também faz codificação e decodificação. Contudo, na Gigabit Ethernet, o transceptor não pode ser externo, pois a GMII não proporciona mecanismos de conexão externa. Examinaremos os transceptores para cada tipo de implementação na próxima seção.

MDI

Exatamente como na Fast Ethernet, a MDI conecta o transceptor ao meio. Para Gigabit Ethernet estão definidos somente o conector RJ-45 e os conectores de fibra óptica.

Implementação da Camada Física

As implementações Gigabit Ethernet, assim como as Fast Ethernet, podem ser classificadas de acordo com a forma de conexão: dois ou quatro pares. A implementação a dois pares é denominada **1000Base-X**, onde estão incluídos a fibra óptica de comprimento de onda curto (**1000Base-SX**) ou comprimento de onda longo (**1000Base-LX**), ou ainda, único meio de cobre STP (**1000Base-CX**). A quarta versão utiliza cabo par trançado (**1000Base-T**). Enfim, são quatro as implementações possíveis da Gigabit Ethernet, conforme mostra a Figura 14.30.

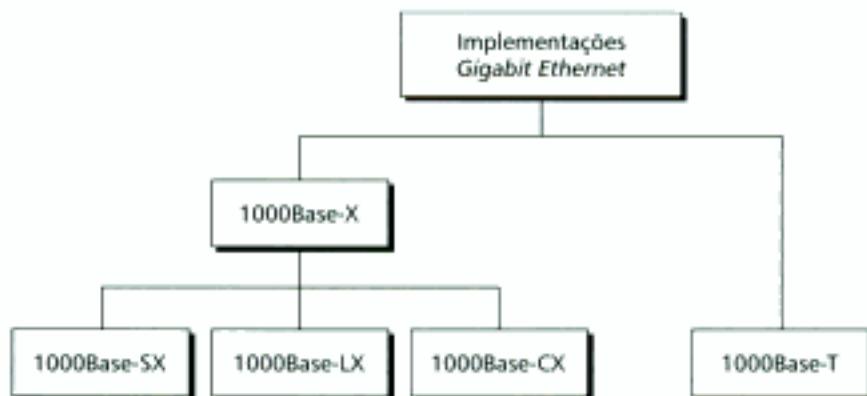


Figura 14.30 Implementações Gigabit Ethernet.

1000Base-X

Tanto o padrão 1000Base-SX quanto o 1000Base-LX usam cabo de fibra óptica como meio de transmissão. A única diferença entre eles é que o primeiro usa *laser* de comprimento de onda curto e o segundo laser de comprimento de onda longo. Como dissemos antes, todas implementações foram projetadas utilizando transceptor interno, sendo impossível a conexão a cabo ou conector na subcamada GMII. A Figura 14.31 ilustra a conexão de um estação ao *hub*.

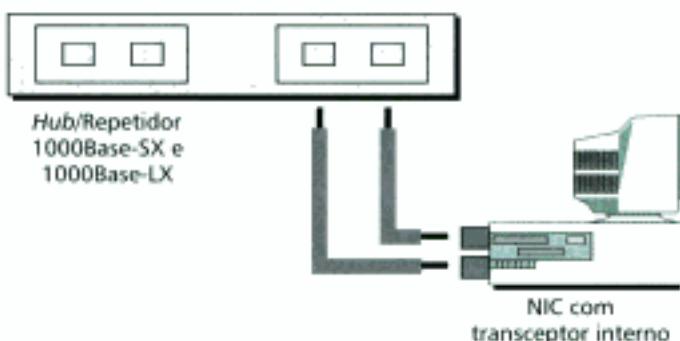


Figura 14.31 Implementação 1000Base-X.

A implementação 1000Base-CX foi desenvolvida para utilizar cabo STP, mas nunca foi implementada na prática.

Transceptor O transceptor na Gigabit Ethernet é interno. Dentre as funções incluem codificação, decodificação, transmissão, recepção e detecção de colisão (caso necessário).

Codificação Para atingir a taxa de dados de 1000 Mbps, o esquema de codificação (e decodificação) acontece em duas etapas, conforme Figura 14.32.

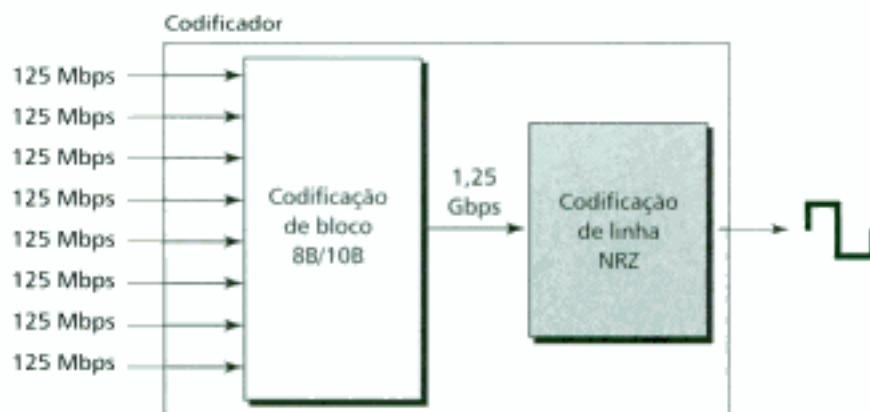


Figura 14.32 Codificação no padrão 1000Base-X.

Para manter o nível de sincronização, o codificador inicialmente realiza um esquema de codificação de blocos. Os 8-bits recebidos paralelamente da interface NIC são codificados num padrão serial de 10 bits através do esquema 8B/10B. Isto requer uma banda de 1,25 GHz (1,25 Gbps).

Em seguida, os dados a 1,25 Gbps são codificados em sinal com codificação NRZ, conforme analisado no Capítulo 4.

1000Base-T

O padrão 1000Base-T foi desenvolvido para utilizar UTP categoria 5 (UTP CAT 5). Os quatro pares trançados juntos atingem uma taxa de 1 Gbps. A Figura 14.33 ilustra a conexão de uma estação ao meio nesta implementação.

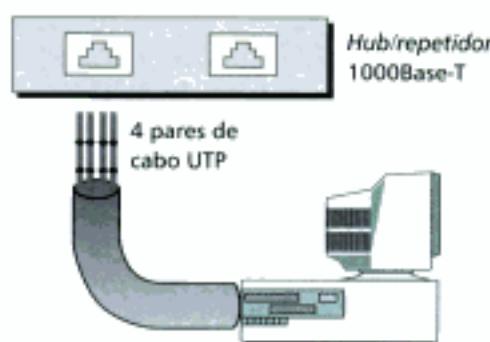


Figura 14.33 Implementação 1000Base-T.

Transceptor Para transmitir a 1,25 Gbps nos quatro pares do cabo UTP, o padrão 1000Base-T usa um esquema de codificação denominado **4D-PAM5** (**4-Dimensional, 5-Level Pulse Amplitude Modulation**). É utilizada modulação PAM em cinco níveis (PAM nível 5). A técnica não será detalhada aqui porque foge ao escopo deste livro. A Figura 14.34 mostra o conceito em linhas gerais.

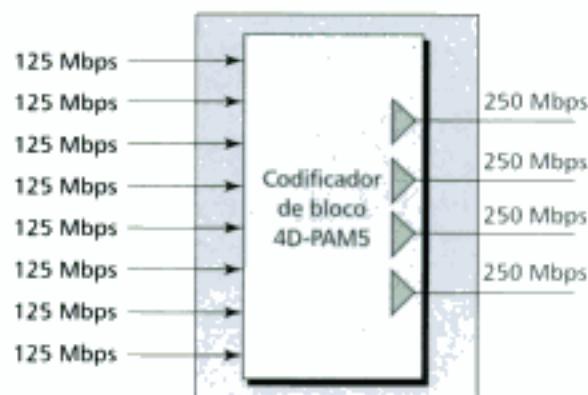


Figura 14.34 Codificação no padrão 1000Base-T.

14.4 TERMOS-CHAVE

1000Base-CX	10Base5
1000Base-LX	10Base-FL
1000Base-SX	10Base-T
1000Base-T	4-Dimensional, 5-Level Pulse Amplitude Modulation (4D-PAM5)
100Base-FX	Attachment Unit Interface (AUI)
100Base-T4	Autonegotiação
100Base-TX	Bridge
100Base-X	
10Base2	

Campo delimitador de início de quadro (<i>Start Frame Delimiter</i> – SFD)	Medium Attachment Unit (MAU)
Domínio de colisão	Medium-Dependent Interface (MDI)
Endereço de broadcast	Medium-Independent Interface (MII)
Endereço de destino (DA)	Network Interface Card (NIC)
Endereço de origem (SA)	Notação Hexadecimal
Endereço multicast	Preâmbulo
Endereço unicast	Subcamada de Controle de Acesso ao Meio (<i>Medium Access Control sublayer</i>)
Ethernet	Subcamada de reconciliação (RS)
Ethernet par trançado	Subcamada PHY
Fast Ethernet	Subcamada Physical Layer Signaling (PLS)
Fiber Link Ethernet	Switched Ethernet
Full-Duplex Switched Ethernet	Thick Ethernet
Gigabit Ethernet	Thin Ethernet
Gigabit Medium-Independent Interface (GMII)	Transceptor

14.5 RESUMO

- Ethernet é o padrão mais difundido para redes locais. Ethernet é quase sinônimo de LAN.
- O padrão IEEE 802.3 define o CSMA/CD 1-persistent como método de acesso para a primeira geração da Ethernet (10 Mbps).
- A camada de enlace da Ethernet é dividida nas subcamadas LLC e MAC.
- A subcamada MAC é responsável pelo funcionamento do método de acesso CSMA/CD.
- Cada estação numa rede Ethernet padrão possui um único endereço de 48-bits gravado no adaptador de rede (Network Interface Card – NIC).
- O tamanho mínimo de um frame na Ethernet de 10 Mbps é 64 bytes. O tamanho máximo é 1518 bytes.
- A camada física da rede Ethernet padrão pode ser composta de quatro subcamadas: Physical Layer Signaling (PLS), Attachment Unit Interface (AUI), Medium Attachment Unit (MAU), Medium-Dependent Interface (MDI).
- A Ethernet padrão (10 Mbps) define quatro diferentes tipos de implementação em banda base: 10Base5 (Thick Ethernet), 10Base2 (Thin Ethernet), 10Base-T (Ethernet par trançado) e 10Base-FL (Fiber Link Ethernet).
- A implementação Ethernet 10Base5 usa cabo coaxial grosso (*thick*). Já a implementação 10Base2 usa cabo coaxial fino (*thin*). Por sua vez, a Ethernet 10Base-T usa par trançado para conectar as estações ao concentrador (*hub*). A implementação 10Base-FL utiliza cabo de fibra óptica.
- Uma bridge pode aumentar a banda de um segmento de rede ou quebrar domínios de colisão numa LAN Ethernet.
- O switch numa LAN Ethernet entrega toda a banda de um segmento de rede à uma estação.
- O modo full-duplex duplica a capacidade de transmissão de cada domínio e acaba com a necessidade do método CSMA/CD.
- O padrão Fast Ethernet transmite a 100 Mbps.
- Numa rede Fast Ethernet, o mecanismo de autonegotiação permite a dois dispositivos negociar o modo ou taxa de transferência de dados.
- A subcamada RS do padrão Fast Ethernet é responsável pela transmissão paralela de dados em 4-bits para a subcamada MII.
- A subcamada MII das redes Fast Ethernet é uma interface utilizada tanto em 10 quanto em 100 Mbps.
- A subcamada PHY das redes Fast Ethernet é responsável pela codificação e decodificação de dados.
- As implementações Fast Ethernet típicas são: 100Base-TX (dois pares de cabo par trançado), 100Base-FX (dois cabos de fibra óptica) e 100Base-T4 (os quatro pares do cabo par trançado).
- O padrão Gigabit Ethernet transmite a 1000 Mbps.
- A Gigabit Ethernet tem dois métodos de acesso: half-duplex com CSMA/CD (pouco utilizado) e full-duplex sem CSMA/CD (padrão de facto).
- A subcamada RS Gigabit Ethernet é responsável pela transmissão paralela de dados em 8-bits para a subcamada PHY via interface GMII.

- A subcamada GMII das redes Gigabit Ethernet define como a subcamada RS será conectada à subcamada PHY.
- A subcamada PHY das redes Gigabit Ethernet é responsável pela codificação e decodificação de dados.
- As implementações Gigabit Ethernet típicas são: 1000Base-SX (duas fibras ópticas e *laser* de comprimento de onda curto), 1000Base-LX (duas fibras ópticas e *laser* de comprimento de onda longo) e 1000Base-T (os quatro pares do cabo par trançado).

14.6 PRATIQUE OS CONHECIMENTOS ADQUIRIDOS

Questões de Revisão

1. Qual é a diferença entre os campos préâmbulo e SFD?
2. Qual é a finalidade de um NIC?
3. Qual é a função de um transceptor?
4. Qual é a diferença entre endereço *multicast* e endereço *broadcast*?
5. Quais são as vantagens de se utilizar *bridges* para segmentar uma rede LAN Ethernet?
6. Qual é a relação entre *switch* e *bridge*?
7. Por que as redes LAN Ethernet operando em modo *full-duplex* não utilizam o método CSMA/CD?
8. Compare as taxas de transmissão das redes Ethernet Padrão, Fast Ethernet e Gigabit Ethernet.
9. Quais são as implementações mais comuns das redes Ethernet padrão?
10. Quais são as implementações mais comuns das redes Fast Ethernet?
11. Quais são as implementações mais comuns das redes Gigabit Ethernet?
12. Qual é a finalidade do mecanismo de auto-negociação?
13. Compare a subcamada RS da rede Fast Ethernet com a subcamada PLS da rede Ethernet padrão.
14. O que é GMII na rede Gigabit Ethernet?
15. Quais camadas do modelo TCP/IP (arquitetura da Internet) se relacionam com as LANs?

Questões de Múltipla Escolha

16. Qual é o equivalente hexadecimal do endereço físico 01011010 00010001 01010101 00011000 10101010 00001111?
 - a. 5A-88-AA-18-55-F0
 - b. 5A-81-BA-81-AA-0F
 - c. 5A-18-5A-18-55-0F
 - d. 5A-11-55-18-AA-0F
17. Se o endereço MAC de origem é 07-01-02-03-04-05, então este é um endereço
 - a. *Unicast*
 - b. *Multicast*
 - c. *Broadcast*
 - d. Nenhuma das respostas
18. Se o endereço MAC de destino é 08-07-06-05-44-33, então este é um endereço
 - a. *Unicast*
 - b. *Multicast*
 - c. *Broadcast*
 - d. Nenhuma das respostas
19. Qual dos seguintes endereços abaixo não pode ser um endereço MAC de origem?
 - a. 8A-7B-6C-DE-10-00
 - b. EE-AA-C1-23-45-32
 - c. 46-56-21-1A-DE-F4
 - d. 8B-32-21-21-4D-34
20. Qual dos seguintes endereços abaixo não pode ser um endereço MAC *unicast*?
 - a. 43-7B-6C-DE-10-00
 - b. 44-AA-C1-23-45-32
 - c. 46-56-21-1A-DE-F4
 - d. 48-32-21-21-4D-34
21. Qual dos seguintes endereços abaixo não pode ser um endereço MAC *multicast*?
 - a. B7-7B-6C-DE-10-00
 - b. 7B-AA-C1-23-45-32
 - c. 4C-56-21-1A-DE-F4
 - d. 83-32-21-21-4D-34
22. Uma rede LAN Ethernet padrão com 10 estações usa uma *bridge* de _____ portas e a taxa média efetiva de transmissão de cada estação vale 2 Mbps.
 - a. 1
 - b. 2
 - c. 5
 - d. 10

23. Uma rede LAN Ethernet com _____ estações usa uma *bridge* de quatro portas. Cada estação possui uma taxa média efetiva de transmissão de dados de 1,25 Mbps.
- 32
 - 40
 - 80
 - 100
24. Quarenta estações estão conectadas em rede LAN Ethernet. Uma *bridge* de 10 portas segmenta a LAN. Qual é a taxa de dados média de cada estação?
- 1,0 Mbps
 - 2,0 Mbps
 - 2,5 Mbps
 - 5,0 Mbps
25. Uma rede Ethernet padrão com 80 estações é quebrada em quatro domínios de colisão. Isto significa que um máximo de _____ estações disputam o acesso ao meio em qualquer instante.
- 320
 - 80
 - 76
 - 20
26. Qual é a eficiência da codificação de blocos 4B/5B?
- 20%
 - 40%
 - 60%
 - 80%
27. Qual é a eficiência de um *frame* transportando um *payload* (carga) de 46 bytes numa rede Gigabit Ethernet?
- 97%
 - 70%
 - 56%
 - 12%
28. Qual das implementações Gigabit Ethernet abaixo transmite a quatro fios?
- 1000Base-SX
 - 1000Base-LX
 - 1000Base-CX
 - 1000Base-T
29. Qual é a eficiência da codificação 8B/10B?
- 20%
 - 40%
 - 60%
 - 80%

Exercícios

30. Qual é o tamanho médio de um *frame* Ethernet padrão?
31. Considerando todo o *frame*, que percentual é reservado a dados (*payload*) no menor *frame* Ethernet? Qual é o percentual para o maior *frame* Ethernet? Qual é o percentual médio?
32. Por que o campo de dados do *frame* Ethernet deve possuir um tamanho mínimo?
33. Imagine que o comprimento de um cabo 10Base5 seja 2500 m. Se a velocidade de propagação do sinal no cabo coaxial grosso vale 200.000.000 m/s, quanto tempo leva para um bit viajar de uma ponta a outra desse cabo? Ignore os atrasos de propagação produzidos no equipamento.
34. A taxa de dados do padrão 10Base5 vale 10 Mbps. Quanto tempo um dispositivo leva para montar o menor *frame* nessa rede? Mostre seus cálculos.
35. A subcamada MAC recebe 42 bytes de dados da subcamada LLC. Quantos bytes de *padding* (enchimento) devem ser adicionados aos dados?
36. A subcamada MAC recebe 1510 bytes de dados da subcamada LLC. Os dados podem ser encapsulados num único *frame*? Caso não, quantos *frames* devem ser montados para transmitir esses dados? Qual é o tamanho do campo de dados em cada *frame*?
37. Complete a Tabela 14.1.
38. Usando a Tabela 14.2, compare as subcamadas físicas dos padrões Fast e Gigabit Ethernet.
39. Usando a Tabela 14.3, compare os diferentes tipos de implementação Fast Ethernet.
40. Usando a Tabela 14.4, compare os diferentes tipos de implementação Gigabit Ethernet.

TABELA 14.1 Exercício 37

<i>Características</i>	<i>10Base5</i>	<i>10Base2</i>	<i>10BaseT</i>	<i>10Base-FL</i>
Tipo de cabo				
Tipo de transceptor				
Necessidade de terminador				

TABELA 14.2 Exercício 38

<i>Subcamadas</i>	<i>Fast Ethernet</i>	<i>Gigabit Ethernet</i>
Reconciliação (RS)		
MII		
GI		
PHY		
MDI		

TABELA 14.3 Exercício 39

<i>Implementação</i>	<i>Meio</i>	<i>Método de codificação</i>
100Base-TX		
100Base-FX		
100Base-T4		

TABELA 14.4 Exercício 40

<i>Implementação</i>	<i>Meio</i>	<i>Método de codificação</i>
1000Base-SX		
1000Base-LX		
1000Base-CX		
1000Base-T		

Redes LANs Sem Fio

A comunicação sem fios é uma das tecnologias que mais tem crescido nos últimos anos. A demanda pela conexão de dispositivos sem a utilização de cabos aumentou vertiginosamente em todo o mundo. Atualmente, as LANs sem fios são encontradas em campos universitários, escritórios de empresas e em áreas públicas. Nas residências, uma **LAN sem fios (Wireless LAN – WLAN)** pode combinar a mobilidade do usuário, conectividade e velocidade de acesso à Internet.

Neste capítulo, concentraremos o foco em duas tecnologias *wireless* emergentes para LANs: WLANs padrão IEEE 802.11, às vezes denominada Wireless Ethernet, e a Bluetooth, uma tecnologia complexa para conectar WLANs pequenas.

15.1 IEEE 802.11

O grupo de trabalho IEEE publicou as especificações do protocolo segundo o **IEEE 802.11**, onde se encontra o detalhamento das camadas física e de enlace das WLANs. Contudo, antes de discutirmos estas camadas, vamos descrever a arquitetura genérica do protocolo.

Arquitetura

O padrão define dois tipos de serviços: o BBS (*Basic Service Set*) e o ESS (*Extended Service Set*).

BSS

O padrão IEEE 802.11 define o **Basic Service Set (BSS)** como o bloco de construção de uma LAN sem fio. Um BSS é construído a partir de estações fixas ou móveis e, possivelmente, uma estação base central, conhecida como **ponto de acesso (Access Point – AP)**. A Figura 15.1 ilustra dois conjuntos montados seguindo as possibilidades descritas acima.

O serviço BSS sem um AP é uma rede isolada e não pode transmitir dados para outros BSSs. É o que chamamos de *arquitetura ad hoc*. Nesta arquitetura, as estações fazem parte de uma rede sem a necessidade de um AP. Elas simplesmente localizam-se reciprocamente e concordam em fazer parte de um BSS.



Figura 15.1 BSSs.

ESS

O mesmo padrão IEEE 802.11 define o **Extended Service Set (ESS)** sendo formado por dois ou mais BSS interligados por APs. Neste caso, os BSSs são conectados através de um *sistema de distribuição* o qual, geralmente, é uma LAN cabeada. O sistema de distribuição conecta os APs nas BSSs. O padrão IEEE 802.11 não restringe o tipo de sistema de distribuição. Portanto, ele pode seguir qualquer padrão de rede local IEEE, por exemplo, a Ethernet padrão ou *Fast Ethernet*. Perceba que o serviço ESS utiliza dois tipos de estações: móveis e fixas. As estações móveis ficam restritas à área de cobertura dentro de um BSS. As estações fixas são os APs que fazem parte de uma rede local tradicional.

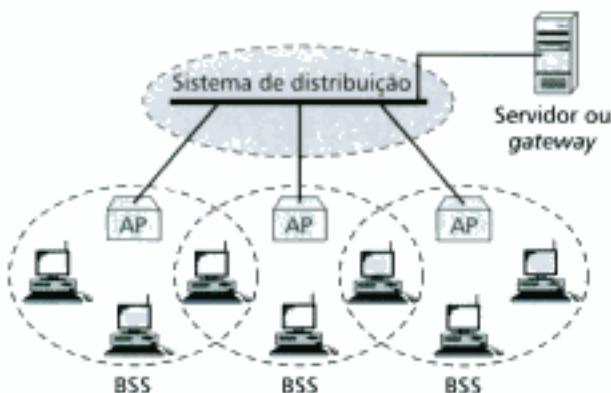


Figura 15.2 ESS.

Quando BSSs estão conectados entre si temos o que é denominada *configuração de infra-estrutura*. Nessa rede, as estações dentro do raio de alcance podem se comunicar sem o suporte do AP. Entretanto, as comunicações entre duas estações em diferentes BSSs usualmente acontecem via dois APs. A idéia é bastante similar à comunicação numa rede de telefonia celular, se considerarmos cada BSS como estação móvel e cada AP como estação base. Perceba ainda que uma estação móvel pode pertencer a mais de um BSS ao mesmo tempo. Basta que ela se localize na região de interseção dos BSS.

Tipos de Estações

O padrão IEEE 802.11 define três tipos de estações, baseado na mobilidade delas dentro da WLAN: **sem transição**, **transição inter-BSS** e **transição inter-ESS**.

Mobilidade Sem Transição Uma estação com mobilidade sem transição é fixa ou se move dentro de um BSS.

Mobilidade Com Transição Inter-BSS Uma estação com mobilidade inter-BSS pode se mover de um BSS para outro, mas o movimento fica confinado dentro de um ESS.

Mobilidade com Transição inter-ESS Uma estação com mobilidade inter-ESS pode se mover de um ESS para outro. Entretanto, o padrão IEEE 802.11 não assegura comunicação contínua durante a transição entre ESS.

Camada Física

O padrão IEEE 802.11 também define as especificações para conversão de *bits* em sinal elétrico na camada física: uma especificação utiliza a faixa de freqüências na região do infravermelho e não será discutida aqui. As outras cinco estão na faixa de rádio freqüências, conforme ilustra a Figura 15.3.



Figura 15.3 Especificações de camada física.

IEEE 802.11 FHSS

O padrão IEEE 802.11 FHSS descreve o método de espalhamento espectral, cujo acrônimo **FHSS** significa **Frequency-Hopping Spread Spectrum**, para geração do sinal na faixa de 2.4 GHz.

FHSS O FHSS é um método onde um transmissor envia uma portadora de freqüência durante um curto intervalo de tempo. Então, salta para outra portadora de freqüência e fica nela durante um mesmo intervalo de tempo. Em seguida, continua saltando até que se repita o ciclo após N saltos (veja a Figura 15.4). Se a largura de banda do sinal original é B , a largura de banda partilhada pelo espectro espalhado (*spread spectrum*) é $N \times B$.

O espalhamento faz com que seja muito difícil pessoas não autorizadas invadirem o sistema para acessarem os dados transmitidos. No FHSS, o transmissor e receptor devem concordar sobre a seqüência de divisão da banda para a manutenção de um único canal lógico. Na figura, o primeiro *bit* ou grupo de *bits* é enviado na faixa 1, o segundo é enviado na faixa 2, e assim por diante. O invasor que conseguir sintonia na freqüência de qualquer faixa pode conseguir receber o primeiro grupo de *bits*, mas não receberá nada dessa faixa durante o segundo intervalo. O intervalo de tempo característico em cada faixa, denominado tempo de habilitação, é tipicamente 400 ms ou mais. Veja que isto não constitui um caso de acesso múltiplo; todas as estações disputam a utilização das mesmas faixas para envio dos respectivos dados. Contenção é o nome da função da subcamada MAC, como veremos adiante.



Figura 15.4 FHSS.

Banda O FHSS usa uma faixa de 2,4 GHz, denominada ISM (*Industrial, Scientific, and Medical*). Ela opera na faixa de 2,4 a 2,48 GHz (dependendo da regulamentação de cada país). Toda a banda é dividida em 79 faixas de 1 MHz cada. Uma função geradora de números pseudo aleatórios seleciona o salto para a faixa habilitada.

Modulação e Taxa de Transmissão A técnica de modulação utilizada nesta especificação é FSK* operando numa taxa de modulação de 1 Mbaud/s. O sistema lida com 1 ou 2-bits/baud (2-FSK ou 4-FSK), o qual resulta numa taxa de transferência de dados de 1 ou 2 Mbps.

IEEE 802.11 DSSS

O padrão IEEE 802.11 DSSS também descreve o método de espalhamento espectral, cujo acrônimo DSSS significa **Direct-Sequence Spread Spectrum**, para geração do sinal na faixa de freqüências ISM próxima a 2,4 GHz.

DSSS No DSSS cada bit enviado pelo transmissor é categoricamente substituído por uma seqüência de bits denominada *chip code* ou *bit-code* (veja Capítulo 13). Porém, para evitar buffering (uso de área de armazenamento temporário) o tempo necessário para transmitir um *chip-code* deve ser o mesmo tempo necessário para transmitir um bit original. Se o número de bits em cada *chip-code* for N , então a taxa de transmissão dos *chip-codes* será N vezes a taxa de transmissão da cadeia original de bits. A Figura 15.5 mostra um exemplo de DSSS.

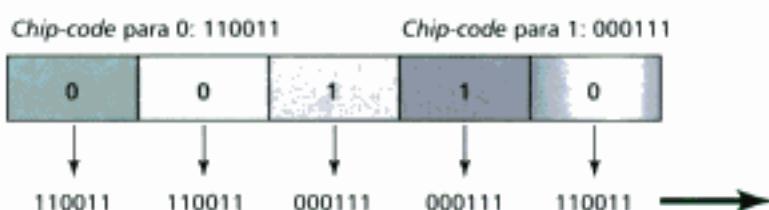


Figura 15.5 DSSS.

Embora este esquema seja similar ao CDMA (veja Capítulo 13) há uma diferença relativa à posição: o DSSS é implementado na camada física. Assim, não se trata de outro método de acesso múltiplo para a camada de enlace. Necessitaremos de um método de contenção na camada de enlace que será discutido adiante.

Banda O DSSS usa uma faixa ISM de 2,4 GHz. A seqüência de bits usa toda a banda.

Modulação e Taxa de Transmissão A técnica de modulação utilizada nesta especificação é PSK operando numa taxa de modulação de 1 Mbaud/s. O sistema lida com 1 ou 2-bits/baud (DBPSK ou DQPSK), o qual resulta numa taxa de transferência de dados de 1 ou 2 Mbps.

IEEE 802.11a OFDM

O padrão IEEE 802.11a OFDM descreve um método de multiplexação, cujo acrônimo **OFDM** significa **Orthogonal Frequency-Division Multiplexing**, para geração do sinal na faixa de freqüências ISM próxima a 5 GHz.

OFDM OFDM é parecido com FDM, com a diferença fundamental de que todas as faixas são utilizadas por uma estação de origem num dado instante de tempo. As estações de origem disputam entre si para acessar a camada de enlace.

Banda A especificação usa uma faixa de freqüências ISM próxima a 5 GHz. Toda a banda é dividida em 52 faixas, onde 48 delas estão destinadas a enviar 48 grupos de bits, num certo de tempo, e as outras 4 são utilizadas para controle da informação. O esquema é muito parecido com ADSL, discutido no Capítulo 9. A divisão da banda em faixas diminui os efeitos da interferência. Além disso, caso as faixas sejam escolhidas aleatoriamente, é possível implementar um nível de segurança maior.

Modulação e Taxa de Transmissão ODFM usa é PSK e QAM para modulação. A transmissão de dados típica ocorre a 18 Mbps (PSK) e 54 Mbps (QAM).

* N. de R. T.: A rigor, a versão de 1 Mbps utiliza 2 níveis da modulação GFSK (*Gaussian Frequency Shift Keying*) e a de 2 Mbps utiliza 4 níveis da mesma modulação.

IEEE 802.11b HR-DSSS

O padrão IEEE 802.11b HR-DSSS descreve um método de espalhamento espectral, cujo acrônimo **HR** significa **High-Rate**, para geração do sinal na faixa de freqüências ISM próxima a 2,4 GHz.

HR-DSSS O HR-DSSS exibe muitas semelhanças com DSSS. A diferença reside no método utilizado para a codificação, o qual é denominado **Complementary Code Keying (CCK)**. O CCK codifica 4 ou 8 bits num único símbolo CCK.

Banda A especificação utiliza uma faixa de freqüência ISM próxima a 2,4 GHz.

Modulação e Taxa de Transmissão Para manter a compatibilidade com DSSS, o HR-DSSS suporta quatro taxas de transmissão: 1; 2; 5,5 e 11 Mbps. As duas primeiras usam a mesma técnica de modulação encontrada no DSSS. A versão de 5,5 Mbps usa DBPSK, modula numa taxa de 1,375 Mbaud/s e codifica com 4-bits por símbolo CCK. A versão de 11 Mbps usa DQPSK, também modula numa taxa de 1,375 Mbaud/s e codifica com 8-bits por símbolo CCK. Perceba que a versão 11 Mbps possui uma taxa de transmissão próxima das redes Ethernet padrão (10 Mbps).

IEEE 802.11g OFDM

Esta especificação é relativamente nova e utiliza OFDM numa faixa de freqüência ISM de 2,4 GHz. Através de uma técnica de modulação complexa, ela atinge a taxa de 54 Mbps.

Subcamada MAC

O padrão 802.11 estabelece duas subcamadas MAC: **Distributed Coordination Function (DCF)** e **Point Coordination Function (PCF)**, conforme ilustra a Figura 15.6.

A subcamada PCF é um método complexo e opcional que pode ser implementado na configuração de infra-estrutura (a rede *ad hoc* não admite a PCF). Não discutiremos isso aqui. Para mais detalhes veja Fourozan, *Local Area Networks*, McGraw-Hill. A subcamada DCF utiliza um método similar ao CSMA/CA, estudado no Capítulo 13, com algumas características de controle adicionais. Analisaremos somente o método de acesso.

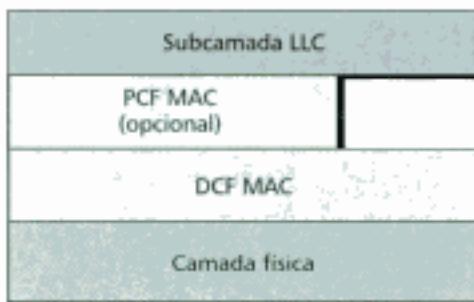


Figura 15.6 Camadas MAC no padrão IEEE 802.11.

CSMA/CA

As LANs sem fio não podem implementar o CSMA/CD por três motivos:

1. Detecção de colisão implica que a estação deve ser capaz de enviar dados e receber sinais de colisão ao mesmo tempo. Isto implica no aumento do custo das estações e aumento dos requerimentos de banda.
2. Uma colisão pode não ser detectada devido a algum tipo de problema relacionado ao fato do dispositivo estar “escondido”. Uma estação pode se apresentar escondida de outra num ambiente *wireless* (devido aos obstáculos naturais, tais como montanhas, ou artificiais, como construções). Por exemplo, suponha que as estações A e B tenham dados para transmitir à estação C. Suponha ainda que a estação B não seja “vista” pela estação A, tal que se ocorrer colisão próximo a estação B, a estação A não tomará conhecimento. Este tipo de situação não ocorre numa LAN cabeada porque todas as estações estão conectadas através de cabos, assim todas as estações são ouvidas pelas demais.

- A distância entre estações numa LAN sem fio pode ser muito grande. Um sinal desvanecido pode impedir uma estação numa extremidade de ouvir a colisão gerada na outra extremidade.

Fluxo do Processo CSMA/CA A Figura 15.7 indica um fluxograma similar ao que encontramos no Capítulo 13. Note que o fluxograma desta figura apresenta algumas modificações em relação ao CSMA/CA do Capítulo 13.

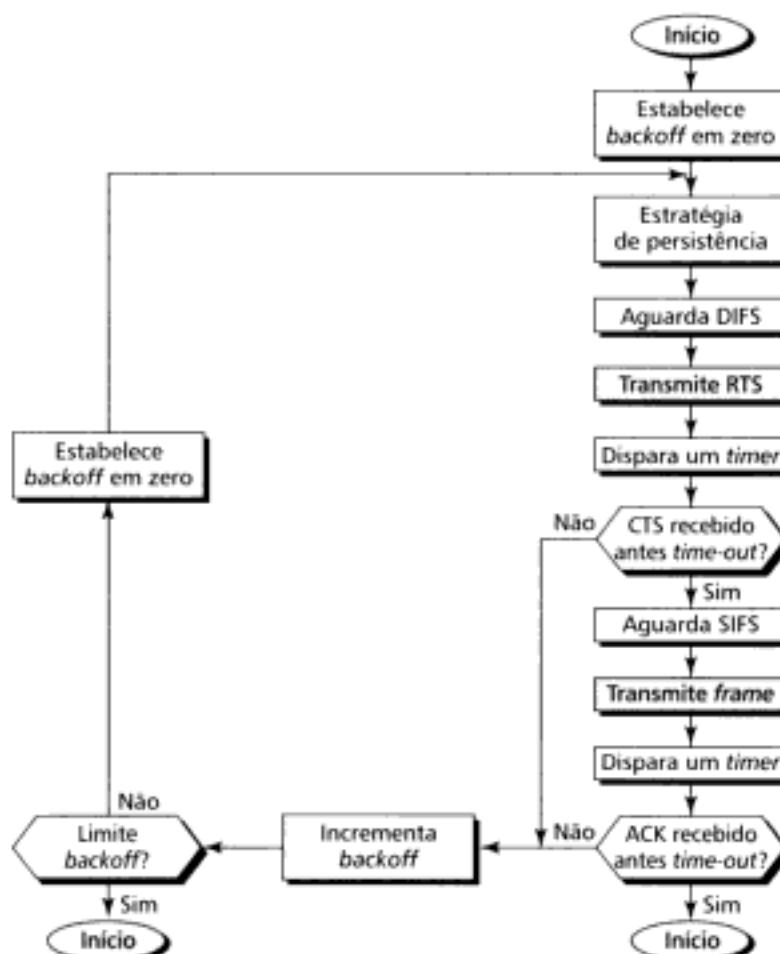


Figura 15.7 Fluxograma CSMA/CA.

Linha de Tempo: Troca de *Frames* A Figura 15.8 mostra a linha de tempo durante a troca de *frames* de dados e de controle.

- Antes de iniciar a transmissão de um *frame*, a estação de origem ouve o meio verificando o nível de energia da portadora de freqüência.
 - O canal utiliza a estratégia de persistência com *backoff* até que o canal fique livre.
 - Após a estação detectar que o meio está livre, ela espera um período de tempo, denominado **Distributed Interframe Space (DIFS)**, para então iniciar a transmissão de um *frame* de controle conhecido como *Request to Send (RTS)*.
- Após receber o RTS e esperar um curto intervalo de tempo, denominado **Short Interframe Space (SIFS)**, a estação de destino também envia um *frame* de controle, conhecido como *Clear to Send (CTS)*, à estação de origem. Esse *frame* indica que a estação de destino está pronta para receber dados.
- Então, após um período de tempo igual ao SIFS, a estação de origem envia os dados.

- Após um período de tempo igual ao SIFS, a estação de destino envia um ACK para mostrar que o frame foi recebido. A confirmação (ACK) é necessária neste protocolo porque a estação de origem não tem como verificar o sucesso do recebimento dos dados no destino. De outro modo, a falta de colisão no CSMA/CA é um modo de indicar à fonte que os dados chegaram ao destino.

Network Allocation Vector Como as outras estações adiam o envio de dados quando uma estação já recebeu acesso ao meio? Noutras palavras, como o aspecto *collision avoidance* deste protocolo é realizado? O elemento chave é denominado NAV.

Quando uma estação envia um frame RTS, ela inclui o tempo de duração necessário para ocupar o canal. As estações que são afetadas pela transmissão criam um relógio de temporização, denominado **Network Allocation Vector (NAV)**, exibindo quanto tempo resta antes das estações olharem se o canal está livre outra vez. Sempre que uma estação acessar o sistema e enviar um frame RTS, as outras estações disparam os respectivos relógios NAV. Noutras palavras, as estações olham primeiramente os relógios NAV para verificar se está liberado o acesso ao meio, antes de verificar se o meio está disponível para transmissão. A Figura 15.8 ilustra toda a idéia do NAV.

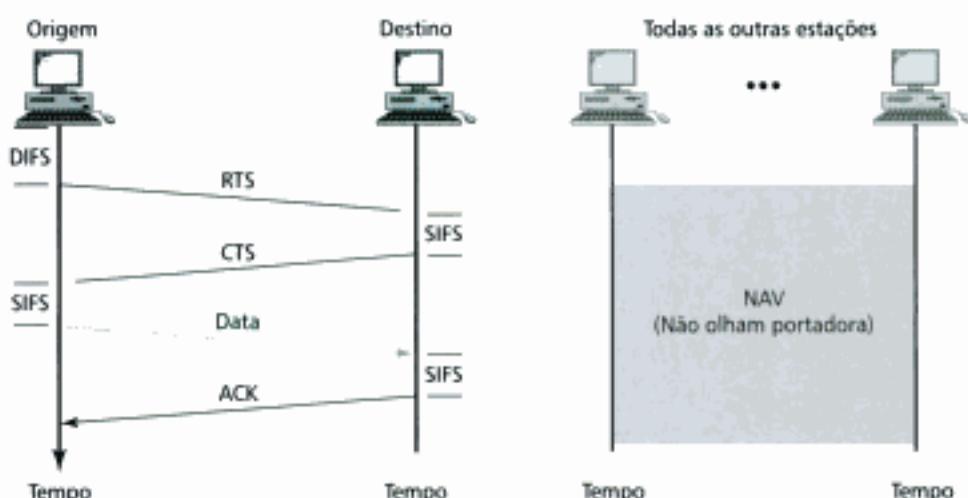


Figura 15.8 CSMA/CA e NAV.

Colisão Durante o Estabelecimento da Comunicação (Handshaking) O que acontece se ocorrer colisão durante o tempo de transição dos frames RTS ou CTS, freqüentemente denominado **período de handshaking**? Por exemplo, duas ou mais estações podem tentar transmitir frames RTS ao mesmo tempo. Neste caso, os frames podem colidir. Entretanto, visto que não existe mecanismos para detecção de colisão, o transmissor assume que ocorreu colisão se um frame CTS não for recebido do receptor. Assim, é utilizada a estratégia de *backoff* e o transmissor tenta transmitir novamente.

Fragmentação

Um ambiente wireless é bastante susceptível a ruídos. Um frame corrompido tem de ser retransmitido. Desse modo, o protocolo recomenda a fragmentação dos frames, ou seja, a divisão de frames maiores em frames menores. A retransmissão tem mais chances de sucesso se ela tentar enviar frames menores.

Formato do Frame

O frame da subcamada MAC possui nove campos, conforme Figura 15.9.

- Campo controle do frame (FC).** O campo FC tem 2 bytes de tamanho e define o tipo de frame, além de trocar informação de controle. A Tabela 15.1 descreve os subcampos. Examinaremos cada tipo de frame neste capítulo.

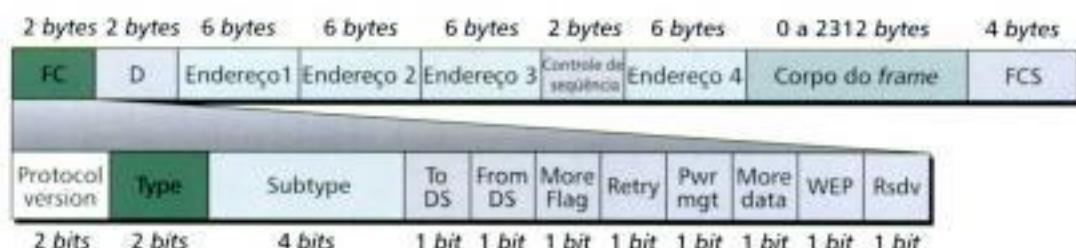


Figura 15.9 Formato do frame.

TABELA 15.1 Subcampos do campo FC

Subcampo	Explicação
Versão	A versão atual do protocolo é a 0
Type	Define o tipo de informação transportada no corpo do frame: gerenciamento (00), controle (01) ou dados (10).
Subtype	Define o subtipo para cada frame (veja a Tabela 15.2).
To DS	Definido mais tarde.
From DS	Definido mais tarde.
More flag	Quando em nível 1 indica mais fragmentos.
Retry	Quando em nível 1 indica frame retransmitido.
Pwr mgt	Quando em nível 1 indica que a estação está no modo de gerenciamento de energia.
More data	Quando em nível 1 indica que a estação tem outros dados para transmitir.
WEP	Wired Equivalent Privacy. Quando em nível 1 indica que a criptografia foi ativada.
Rsvd	Reservado

- **D.** À exceção de um frame, em todos tipos de frames este campo define o tempo de duração da transmissão utilizado no estabelecimento do NAV. Num frame de controle, este campo define a ID (identificação) do frame.
- **Endereços.** Existem quatro campos de endereço, cada qual com 6 bytes de tamanho. O significado de cada campo de endereço depende do valor dos subcampos To DS e From DS. Discutiremos isso adiante.
- **Controle de seqüência.** Este campo define o número seqüencial do frame a ser usado durante o controle de fluxo.
- **Corpo do frame.** Este campo, o qual pode estar entre 0 e 2312 bytes, contém informações baseadas nos subcampos type e subtype do campo FC.
- **FCS.** O campo FCS tem 4 bytes de tamanho e contém uma seqüência de detecção de erro CRC-32.

Tipos de Frames

Uma LAN sem fio definida pelo padrão 802.11 possui três classes de frames: gerenciamento, controle e dados.

Frames de Gerenciamento Estes frames são usados para estabelecer comunicação entre as estações e os pontos de acesso (*access point*).

Frames de Controle Estes frames são utilizados durante o acesso ao canal e para confirmação de frames. A Figura 15.10 mostra o formato.

No caso dos frames de controle o valor do campo type é 01. Os valores possíveis para o campo de subtype são mostrados na Tabela 15.2.

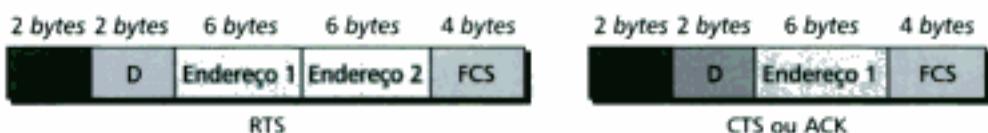


Figura 15.10 Controle de frames.

TABELA 15.2 Valores possíveis para os subcampos dos frames de controle

Subtype	Significado
1011	Request to Send (RTS)
1100	Clear to Send (CLR)
1101	Confirmação (ACK)

Frames de Dados Os frames de dados são utilizados para transportar dados e informação de controle.

Mecanismo de Endereçamento

O mecanismo de endereçamento do padrão IEEE 802.11 é bastante complexo. A complexidade origina-se no fato de que podem existir estações intermediárias (APs) entre origem e destino. Há quatro casos definidos pelo valor dos dois flags no campo FC, isto é, *To DS* e *From DS*. Cada flag pode assumir os estados 0 ou 1, assim, definindo os quatro estados citados acima. A interpretação dos quatro endereços no frame MAC depende do valor desses flags, conforme mostra a Tabela 15.3.

TABELA 15.3 Endereços

To DS	From DS	Endereço 1	Endereço 2	Endereço 3	Endereço 4
0	0	Estação destino	Estação origem	ID do BSS	N/A
0	1	Estação destino	AP transmitindo	Estação origem	N/A
1	0	AP recebendo	Estação origem	Estação destino	N/A
1	1	AP recebendo	AP transmitindo	Estação destino	Estação origem

Perceba que o endereço 1 sempre é o endereço do próximo dispositivo, ou seja, para onde dados ou informação de controle estão indo. O endereço 2 sempre é o endereço da estação de onde vêm dados ou informação de controle. O endereço 3 é o endereço da estação final, caso ele não esteja definido pelo endereço 1. O endereço 4 é o endereço da estação original, caso ele não seja o mesmo do endereço 2.

Caso 1

Este caso acontece quando *To DS* = 0 e *From DS* = 0. Isto significa que o frame não está indo para um sistema de distribuição (*To DS* = 0) e também não está vindo do sistema de distribuição (*From DS* = 0). Logo, o frame só pode estar indo de uma estação para outra num sistema BSS, sem utilizar o sistema de distribuição. O ACK deve ser enviado ao transmissor original. O endereços aparecem ilustrados na Figura 15.11.

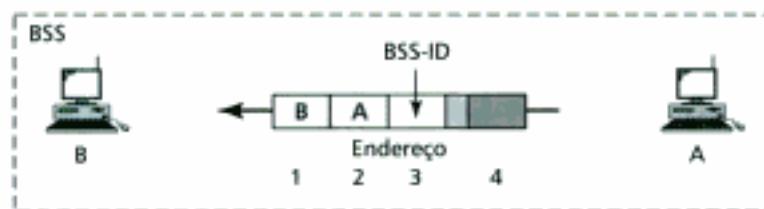


Figura 15.11 Mecanismo de endereçamento: caso 1.

Caso 2

Neste caso $To DS = 0$ e $From DS = 1$. Isto significa que o frame está vindo do sistema de um sistema de distribuição ($From DS = 1$). Assim, o frame está indo de um AP em direção à estação. O ACK deve ser transmitido para o AP. A situação aparece ilustrada na Figura 15.12. Note que o endereço 3 possui o endereço da estação onde se originou o frame (noutra BSS).



Figura 15.12 Mecanismo de endereçamento: caso 2.

Caso 3

Este caso acontece quando $To DS = 1$ e $From DS = 0$. Isto significa que o frame está indo para um sistema de distribuição ($To DS = 1$). Logo, o frame está indo de uma estação para um AP. O ACK é enviado para a estação original. A situação está ilustrada na Figura 15.13. Note que o endereço 3 contém o endereço do destino final do frame (noutra BSS).

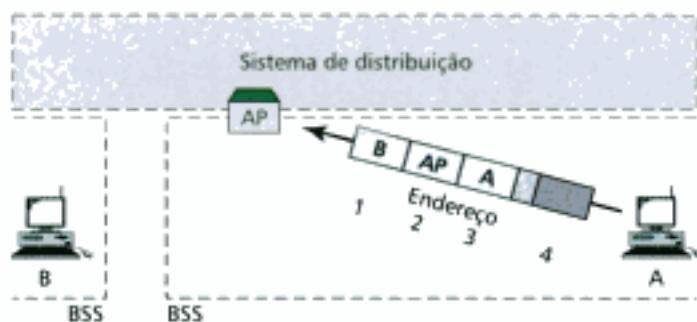


Figura 15.13 Mecanismo de endereçamento: caso 3.

Caso 4

O último caso, $To DS = 1$ e $From DS = 1$, representa a situação na qual o sistema de distribuição também é wireless. O frame segue de um AP para outro formando um sistema de distribuição sem fio. Caso o sistema de distribuição seja uma LAN cabeada não há necessidade de definir endereços porque o frame deve possuir o formato característico da LAN em questão (por exemplo, Ethernet). Entretanto, nesse caso, necessitamos de quatro endereços para definir o transmissor original, o destino final e dois APs intermediários. A Figura 15.14 mostra a situação.

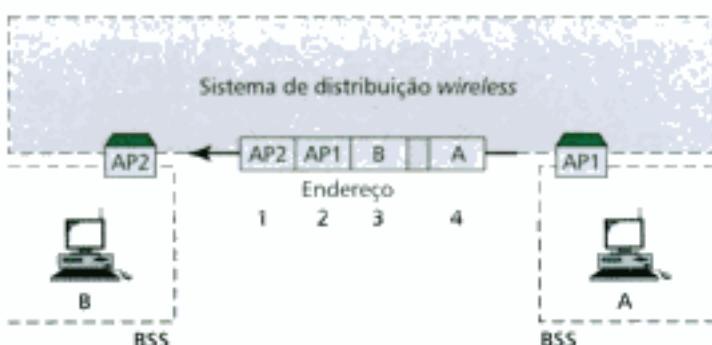


Figura 15.14 Mecanismo de endereçamento: caso 4.

15.2 BLUETOOTH

Bluetooth é o nome da tecnologia WLAN desenvolvida para conectar dispositivos de diferentes funcionalidades como telefones, notebooks, computadores (*desktop* ou *laptop*), câmeras, impressoras, cafeteiras e assim por diante. Uma LAN Bluetooth é uma rede *ad hoc* formada espontaneamente, isto é, os dispositivos, às vezes denominados *gadgets* (equipamentos eletrônicos, em geral, pequenos e modernos), localizam uns aos outros e estabelecem uma rede denominada *piconet*. Uma LAN Bluetooth pode até mesmo se conectar à Internet, se um dos *gadgets* tiver esta capacidade. Por natureza, uma LAN Bluetooth não pode ser grande. Em geral, é um caos quando muitos *gadgets* tentam se conectar à rede.

A tecnologia *Bluetooth* possui muitas aplicações. Atualmente, vários periféricos de computador se comunicam com a CPU através dessa tecnologia (por exemplo, mouse e teclado). Dispositivos de monitoramento podem se comunicar com sensores num pequeno centro de tratamento médico. Dispositivos de segurança residencial podem utilizar esta tecnologia para conectar diferentes sensores a um controlador central de segurança. Os participantes de uma reunião podem sincronizar os respectivos computadores portáteis com o computador do palestrante ou pessoa que conduz a reunião.

Bluetooth foi desenvolvido originalmente pela Ericsson Company. A origem do nome foi uma homenagem ao rei unificador da Dinamarca Harald Blaatand (940-981). *Blaatand* foi traduzido para *Bluetooth* em inglês.

Hoje, a tecnologia *Bluetooth* é a implementação de um protocolo definido pelo padrão IEEE 802.15. O padrão define uma rede sem fio denominada Personal-Area Network (PAN) para operar numa área do tamanho de uma sala. Este escopo de rede gravita em torno do indivíduo. Possui um alcance pequeno, mas efetua a comunicação entre dispositivos pessoais.

Arquitetura

Bluetooth define dois tipos de redes: *piconets* e *scatternet*.

Piconets

Uma rede *Bluetooth* é denominada uma *piconet*. Uma *piconet* pode ter até oito estações, uma das quais é eleita **mestre** e as demais os **escravos**. Todas estações escravas sincronizam os relógios e seqüência de saltos com a estação mestre. Uma *piconet* pode ter uma única estação mestre. A comunicação entre o mestre e os escravos pode ser ponto a ponto ou multiponto. A Figura 15.15 apresenta essa idéia.

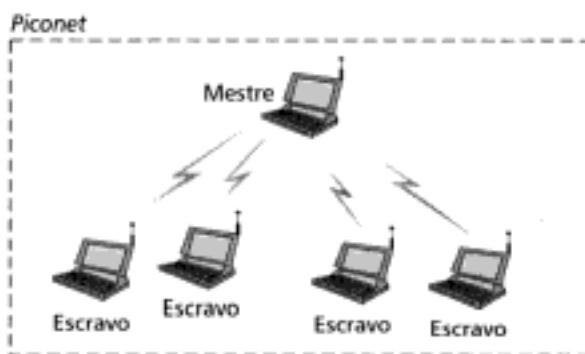


Figura 15.15 Piconet.

Embora uma *piconet* possa ter um máximo de sete escravos é possível adicionar um oitavo no estado estacionado (*parked state*). Um escravo no estado estacionado fica sincronizado com o mestre, mas não pode tomar parte na comunicação até ser movido do estacionamento. Já que somente oito estações podem estar ativas num *piconet*, retirar uma estação do estado estacionado significa levar uma das estações ativas para o estado estacionado.

Scatternet

As piconets podem ser combinadas de modo a formar uma **scatternet**. Uma estação escrava numa piconet pode tornar-se a mestre noutra piconet. Esta estação pode receber mensagens do mestre na primeira piconet (como escrava) e agir como mestre repassando-as às escravas na segunda piconet. Um estação pode pertencer simultaneamente a duas piconets. A Figura 15.16 ilustra uma scatternet.

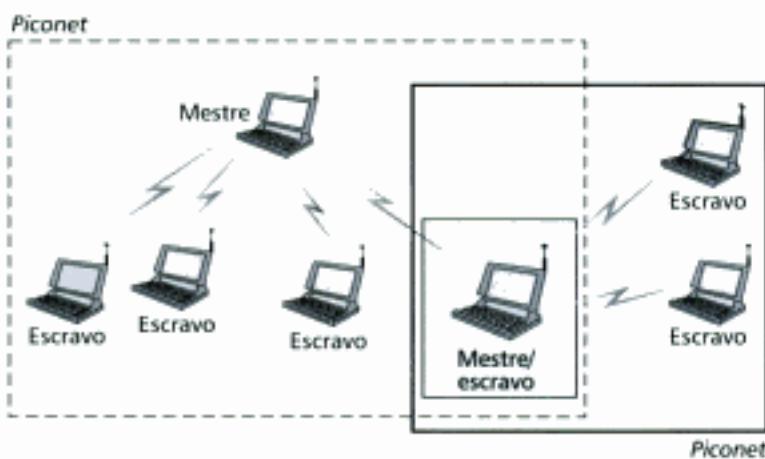


Figura 15.16 Scatternet.

Dispositivos Bluetooth

Todo dispositivo *Bluetooth* possui um transmissor de rádio freqüências. A taxa atual de transmissão de dados é 1 Mbps operando na faixa ISM centrada em 2,4 GHz. Significa que existe a possibilidade de interferência entre as LANs sem fio padrão IEEE 802.11b e as *Bluetooth* LANs.

Camadas Bluetooth

Bluetooth utiliza muitas camadas que não concordam necessariamente com o modelo da Internet definido no primeiro capítulo deste livro. A Figura 15.17 mostra o modelo de camadas *Bluetooth*.

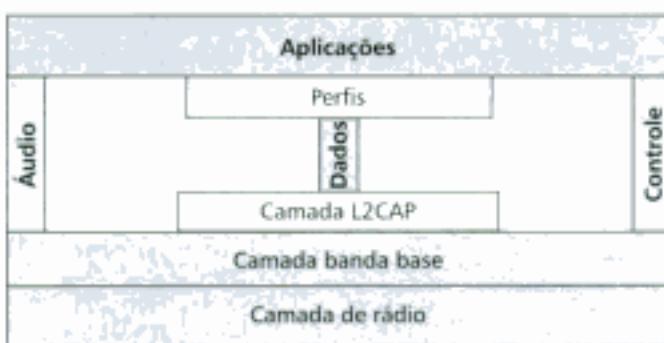


Figura 15.17 Camadas Bluetooth.

Camada de Rádio

Grosso modo, a camada de rádio equivale à camada física do modelo da Internet. Dispositivos *Bluetooth* são dispositivos de baixa potência e têm alcance de 10 m.

Banda

Como dissemos, o *Bluetooth* opera numa faixa ISM centrada em 2,4 GHz dividida em 79 canais espaçados de 1 MHz.

FHSS

Bluetooth utiliza o método FHSS, definido anteriormente, na camada física para evitar interferência de outros dispositivos ou redes. O mestre *Bluetooth* salta 1600 vezes por segundo, isto é, cada dispositivo muda a freqüência de modulação 1600 vezes por segundo. Um dispositivo usa uma freqüência de transmissão durante somente 625 µs (1/1600 s) antes de saltar para a próxima. Assim, o tempo de habilitação é 625 µs.

Modulação

A tecnologia *Bluetooth* usa uma versão sofisticada da modulação FSK para transformar *bits* em sinal eletromagnético. Esta modulação é denominada GFSK (Gaussian FSK), mas não temos a intenção de discuti-la nesse livro. A técnica FSK possui uma freqüência portadora. O *bit* 1 é representado por um deslocamento de freqüência acima da portadora, enquanto o *bit* 0 é representado por um deslocamento de freqüência abaixo da portadora. As freqüências portadoras, em MHz, são definidas de acordo com a seguinte fórmula para cada canal:

$$f_c = 2402 + n \quad n = 0, 1, 2, 3, \dots, 78$$

Por exemplo, o primeiro canal usa portadora em 2402 MHz (2,042 GHz), o segundo canal usa portadora em 2403 MHz (2,403 GHz) e assim por diante.

Camada Banda Base

Esta camada é o equivalente à subcamada MAC nas LANs. O método de acesso utilizado é TDMA (veja Capítulo 13). O mestre e o escravo se comunicam usando *time-slots*. O tamanho de um *time-slot* é exatamente igual ao tempo de habilitação do canal, ou seja, 625 µs. Isto quer dizer, no intervalo de tempo que uma freqüência é utilizada, o mestre envia um *frame* para o escravo ou vice-versa. Note, porém, que a comunicação acontece somente entre o mestre e o escravo, pois os escravos não podem se comunicar uns com os outros.

TDMA

Conforme mencionado, a tecnologia *Bluetooth* utiliza uma forma de TDMA denominada **TDD-TDMA (Time-Division Duplexing TDMA)**. O método TDD é um tipo de comunicação *half-duplex* na qual o escravo e o receptor enviam e recebem dados, mas nunca ao mesmo tempo (*half-duplex*). Entretanto, a comunicação em cada direção utiliza saltos diferentes. Isto lembra os *walkie-talkies*, só que utilizando diferentes freqüências portadoras.

Comunicação Mestre-Escravo Se a *piconet* possuir somente um escravo, a operação TDMA é muito simples. O intervalo de tempo é dividido em *slots* de 625 µs. O mestre usa *slots* de numeração par (0, 2, 4, ...) e o escravo, os *slots* de numeração ímpar (1, 3, 5, ...). O TDD-TDMA permite a comunicação *half-duplex* entre mestre e escravo. No *slot* 0, o mestre envia e o escravo recebe. No *slot* 1, o escravo envia e o mestre recebe. Assim o ciclo vai sendo repetido. A Figura 15.18 apresenta o conceito.

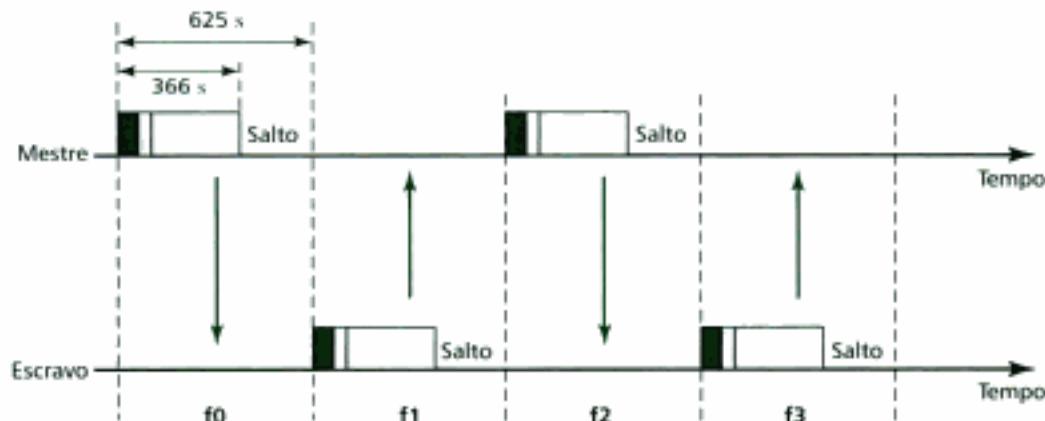


Figura 15.18 Comunicação mestre-escravo.

Comunicação Mestre-Escravos O processo é um pouco mais complexo quando a piconet é formada por mais de um escravo. Novamente, o mestre utiliza *slots* de numeração par, mas um escravo pode transmitir no próximo *slot* com numeração ímpar, se o *frame* no *slot* anterior estava endereçado a ele. Sendo assim, todos os escravos ouvem nos *slots* de numeração par, mas um único escravo pode transmitir em qualquer *slot* com numeração ímpar. A Figura 15.19 mostra o cenário.

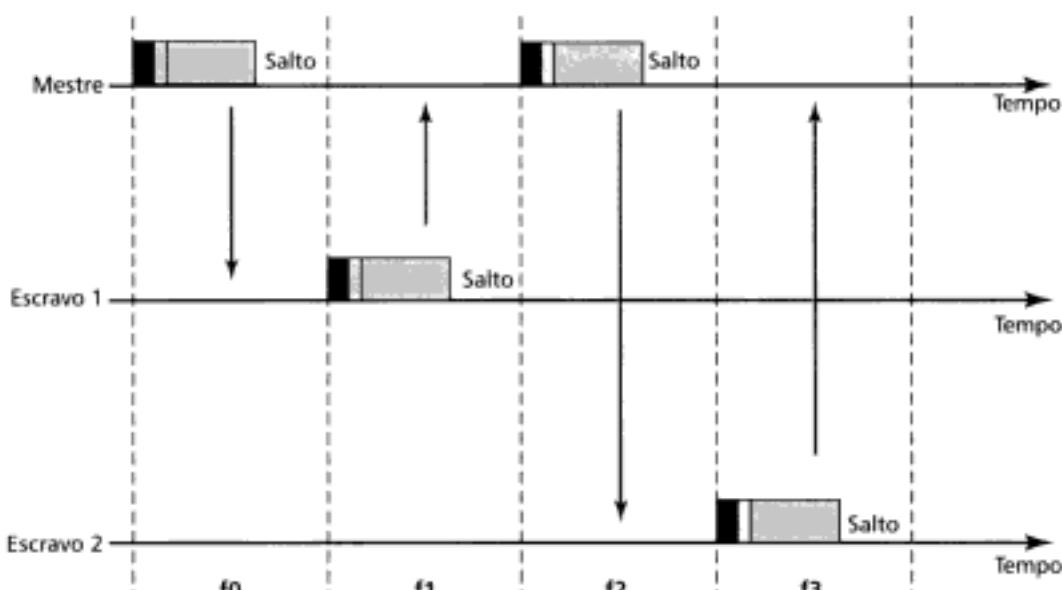


Figura 15.19 Comunicação mestre-escravos.

Vamos sintetizar o processo olhando a figura.

1. No slot 0, o mestre transmite um *frame* para o escravo 1.
2. No slot 1, somente o escravo 1 transmite um *frame* ao mestre porque o *frame* do slot anterior foi endereçado a ele. Os demais escravos permanecem em "silêncio".
3. No slot 2, o mestre transmite um *frame* ao escravo 2.
4. No slot 3, somente o escravo 2 transmite um *frame* ao mestre porque o *frame* do slot anterior foi endereçado a ele. Os demais escravos permanecem em "silêncio".
5. O ciclo continua.

Podemos dizer que este método de acesso é semelhante à operação *polling/selecting* com reserva (veja Capítulo 13). Quando o mestre seleciona um escravo, o mestre também faz um *polling* no escravo. O próximo *slot* é reservado à estação (o escravo) que recebeu o *polling* para que ela transmita seu *frame*. Se a estação não tiver *frames* a transmitir, o canal é mantido em silêncio.

Links Físicos

Existem dois tipos de *links* que podem ser criados entre um mestre e um escravo: *link SCO* e *link ACL*.

SCO Um *link SCO (Synchronous Connection-Oriented)* é utilizado sempre que a latência (atraso na entrega dos dados) for mais importante que a integridade (entrega livre de erros). No SCO, um *link* físico é criado entre um mestre e um escravo reservando *slots* específicos em intervalos regulares. A unidade básica de conexão são dois *slots*, um para cada direção. Além disso, se um *frame* for danificado ele nunca é retransmitido no método SCO. Este tipo de conexão física é bastante utilizada nas aplicações de áudio em tempo real onde a baixa latência é que importa. Por fim, um escravo pode estabelecer até três *links SCO* com o mestre, transmitindo áudio digitalizado a 64 kbps (PCM) em cada *link*.

ACL Um link **ACL (Asynchronous Connectionless Link)** é utilizado quando a integridade dos dados é mais importante que a latência. Neste tipo de *link*, se um *payload* encapsulado no *frame* for corrompido, ele é retransmitido. Um escravo retorna um *frame* ACL no primeiro slot com numeração ímpar disponível se, e somente se, o *slot* endereçado anteriormente era o dele. O *link* ACL pode utilizar um, três ou mais *slots* e pode chegar a uma taxa de transmissão de 721 kbps.

Formato do Frame

Um *frame* da camada banda base pode ser: 1-slot, 3-slots ou 5-slots. Um *slot*, como dissemos antes, tem 625 µs. Entretanto, na troca do *frame* 1-slot, são necessários 259 µs aos mecanismos de controle e de saltos. Isto significa que o *frame* 1-slot tem somente $625 - 259 = 366$ µs. Utilizando uma largura de banda de 1 MHz, o tamanho do *frame* 1-slot é 366 bits.

O *frame* 3-slots ocupa, é claro, três *slots*. Entretanto, visto que 259 µs são reservados, o tamanho desse *frame* é $3 \times 625 - 259 = 1616$ µs ou 1616 bits. Um dispositivo que utilize o *frame* 3-slots permanece no mesmo salto (na mesma portadora) durante três *slots*. Ainda que seja utilizando somente um salto, três saltos são consumidos. Quer dizer que o número do salto para cada *frame* é igual ao número do salto do primeiro *slot*.

O *frame* 5-slots também reserva 259 bits. Isto significa que o tamanho desse *frame* é $5 \times 625 - 259 = 2866$ µs ou 2866 bits.

A Figura 15.20 mostra o formato para os três tipos de *frames*.

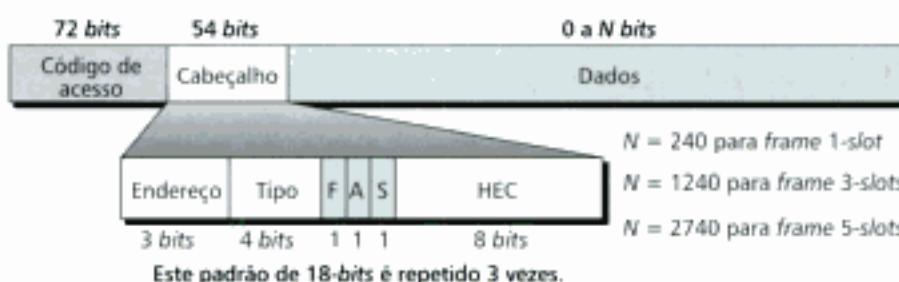


Figura 15.20 Tipos de formato de frames.

Descrevemos a seguir cada campo desses frames:

- **Access Code (código de acesso).** Este campo tem 72-bits e normalmente contém os bits de sincronização e do identificador do mestre (ID) para distinguir um *frame* de diferentes piconets.
- **Header (cabeçalho).** Este campo tem 54-bits, mas é um padrão de 18-bits repetidos três vezes. Cada seção de 18-bits do cabeçalho se divide nos seguintes campos:
 - **Address (endereço).** O campo de address tem 3-bits. Por isso, pode definir até sete escravos (1 a 7). Se o valor desse campo é zero, indica comunicação broadcast do mestre para os escravos.
 - **Type (tipo).** Este campo tem 4-bits e define o tipo de informação que está chegando das camadas superiores. Analisaremos estes tipos mais tarde.
 - **F.** Este campo possui 1 bit para controle de fluxo. Quando estiver em nível 1, indica que o dispositivo está desabilitado a receber mais frames (buffer cheio).
 - **A.** Este campo possui 1 bit para confirmação (ACK). Bluetooth usa Stop-and-wait ARQ, um bit é suficiente para ACK.
 - **S.** O campo S informa o número de seqüência do *frame*. Como Bluetooth utiliza o protocolo Stop-and-wait ARQ, um bit é suficiente para o número de seqüência.

- **HEC.** O cabeçalho possui um campo de correção de erro de 8-bits que faz, basicamente, o *checksum* nos 18-bits de cada seção para detecção de erros.

O cabeçalho possui três seções de 18-bits e o transmissor faz três cópias idênticas dele antes de transmitir. O receptor compara *bit a bit* estas três seções. Na comparação, se os três *bits* forem iguais ele é aceito. Senão, vence o *bit* em quantidade majoritária. Este é um mecanismo direto de correção de erros (para o cabeçalho). É necessário este controle duplo de erros devido à natureza sutil da comunicação (via ar). Note que não existe retransmissão nessa subcamada.

- **Payload.** Este campo tem tamanho que varia de 0 a 2740 bits. Nele, estão contidos dados e/ou informação de controle das camadas superiores.

L2CAP

A subcamada *Bluetooth* equivalente à LLC nas LANs é a **Logical Link Control and Adaptation Protocol (L2CAP)**. Ela é utilizada para troca de dados nos *links ACL*. Os canais SCO não utilizam L2CAP. A Figura 15.21 mostra o formato do pacote de dados neste nível.



Figura 15.21 Formato do pacote de dados L2CAP.

O campo de *length* de 16 bits define (em bytes) o tamanho da seção de *payload*, isto é, os dados oriundos das camadas superiores. Este campo chega até a 65.535 bytes. O campo de identificação do canal (*Channel ID – CID*) identifica univocamente o canal virtual criado neste nível (veja descrição abaixo).

A L2CAP possui muitos serviços específicos: multiplexação, segmentação e reagrupamento, qualidade de serviço (QoS) e gerenciamento de grupo.

Multiplexação

A L2CAP pode fazer multiplexação. No lado do transmissor, ela aceita dados de um dos protocolos das camadas superiores, faz os enquadramentos e os entrega à camada banda base para despacho. No lado do receptor, ela recebe o *frame* na camada banda base, extrai os dados e os entrega ao protocolo apropriado de camada superior. A multiplexação cria uma espécie de canal virtual que será discutido nos capítulos sobre os protocolos das camadas superiores.

Segmentação e Reagrupamento

O tamanho máximo do campo *payload* da camada banda base é 2774 bits (343 bytes). Isto inclui os 4 bytes necessários à definição do tipo e tamanho do pacote. Assim, o maior pacote que pode chegar da camada superior para ser encapsulado nessa camada é somente 339 bytes. Entretanto, às vezes, a camada de aplicação necessita enviar pacotes de dados de até 65.535 bytes (à Internet, por exemplo). A camada L2CAP divide estes pacotes grandes em segmentos menores e adiciona informações extra para definir a localização dos segmentos no pacote original. A L2CAP segmenta os pacotes na origem e os regrupa (remonta) no destino.

QoS

Bluetooth permite que as estações definam o nível de qualidade de serviço (QoS). Discutiremos QoS no Capítulo 23. Por hora, é suficiente saber que, se não estiver definido o nível de qualidade de serviços, o padrão ou *default* do *Bluetooth* é o serviço de melhor esforço (*best-effort*).

Gerenciamento de Grupo

Outra funcionalidade da camada L2CAP é permitir que os dispositivos criem um tipo de endereço lógico entre eles. Isto é bastante similar ao *multicasting*. Por exemplo, dois ou três escravos podem fazer parte de um grupo de *multicast* para receber dados do mestre.

Outras Camadas Superiores

Bluetooth define muitos protocolos das camadas superiores que chamam os serviços da camada L2CAP. Estes protocolos são muito específicos. São também muito complexos e, por isso, resolvemos não abrir espaço para discuti-los aqui.

15.3 TERMOS-CHAVE

Asynchronous Connectionless Link (ACL)	Mestre
Basic Service Set (BSS)	Mobilidade inter-BSS
<i>Bluetooth</i>	Mobilidade inter-ESS
Complementary Code Keying (CCK)	Mobilidade sem transição
Direct Sequence Spread Spectrum (DSSS)	Network Allocation Vector (NAV)
Distributed Coordination Function (DCF)	Orthogonal Frequency-Division Multiplexing (OFDM)
Distributed Interframe Space (DIFS)	Período de <i>handshaking</i>
Escravo	<i>Piconet</i>
Extended Service Set (ESS)	Point Coordination Function (PCF)
Frequency-Hopping Spread Spectrum (FHSS)	Ponto de acesso (<i>Access Point</i> – AP)
High-Rate Direct Sequence Spread Spectrum (HR-DSSS)	<i>Scatternet</i>
IEEE 802.11	Short Interframe Space (SIFS)
LAN sem fio (WLAN)	Synchronous Connection-Oriented link (SCO)
Logical Link Control and Adaptation Protocol (L2CAP)	Time Division Duplexing TDMA (TDD-TDMA)

15.4 RESUMO

- O padrão IEEE 802.11 para LANs sem fio define dois serviços: BSS (Basic Service Set) e Extended Service Set (ESS). Um ESS consiste de dois ou mais BSSs, onde cada BSS deve possuir um ponto de acesso (*Access Point* – AP).
- Os métodos da camada física utilizados pelas WLANs incluem Frequency-Hopping Spread Spectrum (FHSS), Direct Sequence Spread Spectrum (DSSS), Orthogonal Frequency-Division Multiplexing (OFDM) e High-Rate Direct Sequence Spread Spectrum (HR-DSSS).
- FHSS é um método de geração do sinal onde seqüências repetidas das freqüências portadoras são utilizadas para proteção contra *hackers*.
- Um *bit* é substituído por um *chip code* no método DSSS.
- O método OFDM especifica que uma fonte deve usar todos os canais da banda disponível.
- HR-DSSS é uma variação do método DSSS que utiliza um método de codificação denominado Complementary Code Keying (CCK).
- O método de acesso das WLANs é o CS-MAC/CA.
- O Network Allocation Vector (NAV) é um relógio para o *collision avoidance* (impeditimento de congestionamento).
- O *frame* da camada MAC possui nove campos. O mecanismo de endereçamento pode varrer até quatro endereços.
- As WLANs usam *frames* de gerenciamento, controle e de dados.
- *Bluetooth* é uma tecnologia WLAN que conecta dispositivos (denominados *gadgets*) numa mesma área.
- Uma rede *Bluetooth* é chamada de *piconet*. Muitas *piconets* juntas formam uma rede denominada *scatternet*.
- A camada de rádio *Bluetooth* realiza funções semelhantes àquelas encontradas na camada física do modelo da Internet (TCP/IP).
- A camada banda base *Bluetooth* realiza funções semelhantes àquelas encontradas na subcamada MAC.
- Uma rede *Bluetooth* consiste de um dispositivo mestre e cerca de sete dispositivos escravos.
- Um *frame* *Bluetooth* consiste de dados assim como de mecanismos de controle e de saltos. O *frame* 1, 3 e 5-slots tem o tamanho de cada *slot* igual a 625 µs.

15.5 PRATIQUE OS CONHECIMENTOS ADQUIRIDOS

Questões de Revisão

1. Qual é a diferença entre um BSS e um ESS?
2. Discuta os três tipos possíveis de mobilidade dentro das WLANs.
3. O que é FHSS?
4. O que é DSSS?
5. Qual é a diferença entre OFDM e FDM?
6. Qual é o método de acesso utilizado nas WLANs?
7. Qual é a finalidade do relógio NAV?
8. Quais são os três tipos de *frames* utilizados nas redes WLANs?
9. Como um *frame* de controle difere do *frame* de gerenciamento?
10. Cite duas aplicações para as redes Bluetooth.
11. Compare uma *piconet* com uma *scatternet*.
12. Compare o modelo de camadas *Bluetooth* com o modelo da Internet.
13. Quais são os dois tipos de *links* entre um mestre e um escravo de uma rede *Bluetooth*?
14. Na comunicação mestre-escravos, quem utiliza os *slots* de numeração par e os *slots* de numeração ímpar?
15. Durante quanto tempo um *frame* 1-slot numa rede *Bluetooth* utiliza o mecanismo de salto? E os *frames* 3 e 5-slots?
16. Qual é a finalidade da camada L2CAP?

Questões de Múltipla Escolha

17. Uma rede LAN sem fio (WLAN) usando FHSS salta 10 vezes por ciclo. Se a largura de banda do sinal original é 10 MHz, o espectro espalhado (*spread spectrum*) é _____ MHz.
 - 10
 - 100
 - 1000
 - 10.000
18. Uma WLAN utilizando FHSS salta 10 vezes por ciclo. Se a largura de banda do sinal original é 10 MHz e a menor frequência vale 2 GHz, a maior frequência do sistema é _____ GHZ.
 - 1,0
 - 2,0
 - 2,1
 - 3,0
19. Uma WLAN FHSS possui um espectro espalhado de 1 GHz. A largura de banda do sinal original é 250 MHz e o número de saltos é _____ vezes por ciclo.
 - 1
 - 2
 - 3
 - 4
20. Uma WLAN utilizando DSSS e um *chip code* de 8-bits necessita _____ MHz para a transmissão de dados que, originalmente, requeriam uma banda de 10 MHz.
 - 2,5
 - 20
 - 25
 - 40
21. Uma WLAN utilizando DSSS e um *chip code* de 4-bits necessita de 10 MHz para transmissão de dados que, originalmente, requeriam uma banda de 20 MHz.
 - 2
 - 8
 - 16
 - 32
22. Uma WLAN usando DSSS e um *chip code* de 4-bits necessita de 10 MHz para transmissão de dados que, originalmente, requeriam uma banda de _____ MHz.
 - 2,5
 - 20
 - 25
 - 40
23. Numa ESS a estação _____ não é móvel.
 - AP
 - Servidor
 - BSS
 - Nenhuma das anteriores

24. Numa ESS as estações _____ fazem parte de LAN cabeada.
- AP
 - Servidor
 - BSS
 - Todas anteriores
25. Uma estação possuindo mobilidade _____ pode se mover entre as BSS.
- Sem transição
 - Com transição inter-BSS
 - Com transição inter-ESS
 - (b) e (c)
26. Um estação possuindo mobilidade _____ pode se mover entre as ESS.
- Sem transição
 - Com transição inter-BSS
 - Com transição inter-ESS
 - (b) e (c)
27. Uma estação com mobilidade _____ é fixa ou está se movendo dentro de uma BSS.
- Sem transição
 - Transição inter-BSS
 - Transição inter-ESS
 - (a) e (b)
28. Um *frame* _____ normalmente precede um *frame* CTS.
- DIFS
 - SIFS
 - RTS
 - Todas as anteriores
29. Um *frame* _____ normalmente precede um *frame* RTS.
- DIFS
 - CIFS
 - CTS
 - Nenhuma das anteriores
30. As estações não verificam o meio durante o tempo _____.
- RTS
 - CTS
 - SIFS
 - NAV
31. Uma transmissão sem fio é _____ suscetível a erros que uma rede cabeada.
- Mais
 - Menos
 - Meio
 - Nenhuma das anteriores
32. Qual a subcamada MAC que o padrão IEEE 802.11 define?
- LLC
 - PCF
 - DCF
 - (b) e (c)
33. Qual é o método de acesso básico das WLANs, definido no IEEE 802.11?
- LLC
 - DCF
 - PCF
 - BFD
34. O método de acesso básico usado nas WLANs, definido pelo IEEE 802.11, baseia-se no _____.
- CSMA
 - CSMA/CD
 - CSMA/CA
 - Passagem de permissão
35. FHSS, DSSS e OFDM são especificações da camada _____.
- Física
 - De enlace
 - De rede
 - De transporte
36. No método _____, o transmissor salta de frequência em frequência numa determinada ordem.
- FHSS
 - DSSS
 - OFDM
 - HR-DSSS
37. Uma WLAN usa os *frames* de _____ para confirmação.
- Gerenciamento
 - Controle
 - Dados
 - Nenhuma das anteriores
38. Uma WLAN usa os *frames* de _____ para estabelecimento da comunicação entre estações e APs.
- Gerenciamento
 - Controle
 - Dados
 - Nenhuma das anteriores
39. Uma rede *Bluetooth* pode ter _____ mestre(s).
- Um
 - Dois
 - Três
 - Oito

40. As _____ combinadas formam *scatternets*.
- BSSs
 - ESSs
 - APs
 - Piconets
41. Bluetooth usa _____ na camada física.
- FHSS
 - DSSS
42. Um *frame* Bluetooth necessita de _____ μ s para os mecanismos de salto e de controle.
- 625
 - 259
 - 3
 - Um múltiplo de 259

Exercícios

43. Utilizando a Tabela 15.4, compare e contraste os três tipos de mobilidade para uma estação definida no padrão IEEE 802.11.
44. Compare e contraste CSMA/CD com CSMA/CA.
45. Use a Tabela 15.5 para comparar e contrastar nos padrões IEEE 802.3 e IEEE 802.11.

TABELA 15.4 Exercício 43

Tipos de mobilidade	Movimento dentro da BSS	Movimento entre BSSs	Movimento entre ESSs
Sem transição			
Transição inter-BSS			
Transição inter-ESS			

TABELA 15.5 Exercício 45

Campos	Tamanho do campo IEEE 802.3	Tamanho do campo IEEE 802.11
Endereço de destino		
Endereço de origem		
Endereço 1		
Endereço 2		
Endereço 3		
Endereço 4		
FC		
EID		
SC		
Tamanho PDU		
Dados e padding (payload)		
Corpo do frame		
FCS (CRC)		

Interligando LANs, Redes *Backbone* e LANs Virtuais (VLANs)

Normalmente, as redes LANs não constituem redes isoladas. Elas são interconectadasumas às outras formando *internetworks*, como a Internet, por exemplo. Utilizamos dispositivos denominados ativos de rede para conectar LANs ou segmentos de LANs. Tais dispositivos funcionam em diferentes camadas da arquitetura da Internet. Neste capítulo, analisaremos apenas os dispositivos que funcionam nas camadas física e de enlace. No Capítulo 19 voltaremos nosso foco aos dispositivos que operam nas três primeiras camadas da arquitetura da Internet.

Após a análise das funcionalidades de alguns dos dispositivos de redes locais (os ativos de redes), mostraremos como podemos utilizá-los para criar redes *backbone*. Por último, discutiremos as redes locais virtuais ou VLANs.

16.1 ATIVOS DE REDES: DISPOSITIVOS DE REDES LOCAIS

Existem cinco tipos de **ativos de redes**: repetidores, *hubs*, *bridges*, *switches* de camada 2 e 3 e roteador. Repetidores e *hubs* operam somente na primeira camada da arquitetura da Internet. As *bridges* e os *switches* de camada 2 operam nas duas primeiras camadas. Os roteadores e os *switches* de camada 3 funcionam nas três primeiras camadas. A Figura 16.1 associa os dispositivos ativos de redes com as camadas do modelo.

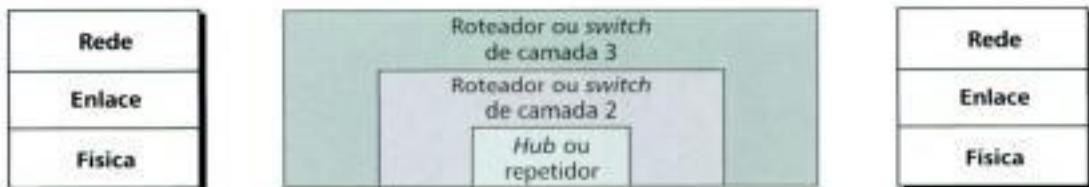


Figura 16.1 Ativos de redes: dispositivos de redes locais.

Repetidores

Um **repetidor** é um dispositivo que opera somente na camada física. Os sinais transportando informação dentro de uma rede só podem viajar distâncias fixas antes que a atenuação coloque em risco a integridade dos dados. Um repetidor recebe um sinal na entrada e, antes que o sinal torne-se muito fraco ou corrompido, regenera-o ao nível onde a inteligibilidade dos dados seja mantida.

O repetidor transmite para a saída um sinal regenerado. A maior aplicação dos repetidores é para estender o comprimento físico de uma LAN, conforme ilustra a Figura 16.2.

Um repetidor não conecta efetivamente duas LANs. De fato, o repetidor conecta dois segmentos de uma mesma LAN. Os segmentos ainda são parte de uma única LAN. Assim, um repetidor não é um dispositivo que conecta duas LANs operando em protocolos diferentes.

Um repetidor conecta segmentos de uma mesma LAN.

Através de um repetidor podemos, por exemplo, estender o limite máximo de uma rede *Ethernet* 10Base5. Nesse padrão, o comprimento máximo de cabo coaxial é 500 m. Para aumentá-lo, dividimos o comprimento máximo desejável de cabeamento em segmentos e instalamos repetidores entre eles. Perceba que a rede como um todo ainda é considerada uma LAN, mas as porções de rede separadas pelos repetidores são denominadas **segmentos***. O repetidor funciona como um nó de duas portas, operando apenas na camada física. Quando o repetidor recebe um *frame* numa das portas, ele regenera o *frame* e o direciona à(s) outra(s) porta(s).

Um repetidor replica todos os *frames* às demais portas; mas não possibilita nenhum tipo de filtragem de *frames*.

É tentador comparar um repetidor a um amplificador, mas a comparação é incorreta. Um **amplificador** não consegue discriminar entre o sinal verdadeiro e o ruído. Ele amplifica com a mesma eficiência tudo que chega à entrada de sinal. Um repetidor não amplifica o sinal, ele regenera-o. Quando o repetidor recebe um sinal atenuado ou corrompido, ele faz uma cópia *bit a bit* do sinal, restabelecendo a intensidade original.

Um repetidor é um regenerador de sinais e não um amplificador.

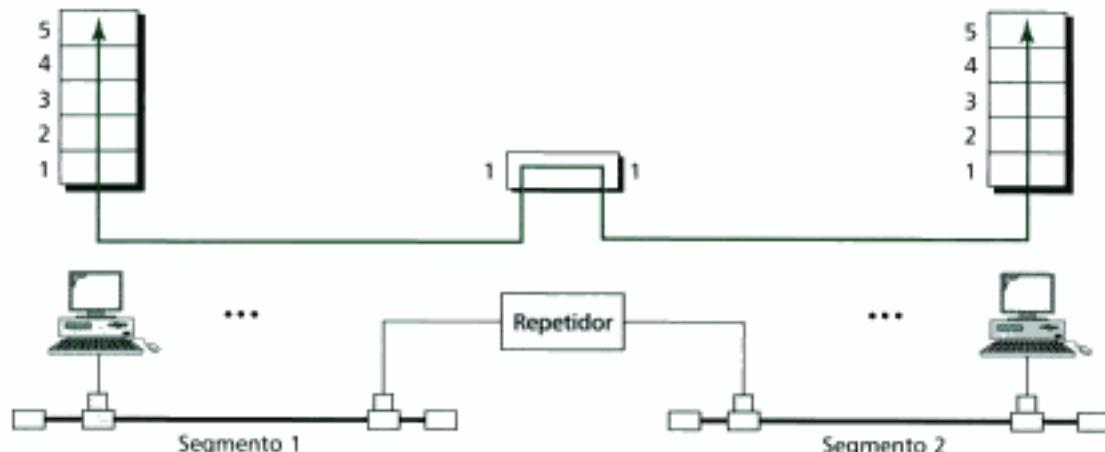


FIGURA 16.2 Repetidor.

* N. de R. T.: Existem regras para segmentação de redes. Por exemplo, a regra utilizada em redes Ethernet padrão é a 5-4-3-2-1. Nessa regra, não podem haver mais de 5 segmentos de rede, 4 repetidores, 3 segmentos contendo máquinas, 2 segmentos sem estações. O número 1 revela que os cinco segmentos juntos perfazem um grande domínio de colisão com no máximo 1024 estações e 2500 m de extensão.

É de suma importância a localização do repetidor num *link*. Ele deve ser colocado na rede de tal maneira que o sinal alcance o repetidor antes que ruídos no *link* destruam a informação (padrão de *bits*) transportada pelo sinal. Níveis relativamente baixos de ruídos podem alterar a precisão dos níveis de tensão que representam os *bits* sem destruir a identidade do sinal (veja a Figura 16.3). Entretanto, se os *bits* corrompidos viajarem em velocidades maiores, ruídos acumulados ao longo do *link* danificam permanentemente o padrão do sinal de dados. Um repetidor bem colocado no *link* pode assegurar a legibilidade do sinal, apesar do sinal ter passado por atenuações, insuficientes para destruir a informação contida no padrão.



Figura 16.3 Função do repetidor.

Hubs

Embora, na acepção geral da palavra, *hub* possa se referir a qualquer dispositivo de conectividade, no âmbito das redes de computadores ele tem um significado bastante específico. Um **hub** é tão somente um repetidor multiportas. Geralmente, o *hub* é utilizado para estabelecer a conexão física entre estações formando uma topologia estrela. Vimos exemplos de *hubs* em algumas implementações *Ethernet* (por exemplo, 10Base-T). Entretanto, *hubs* também podem ser utilizados para criar níveis múltiplos de hierarquia dentro de uma rede, conforme ilustra a Figura 16.4.

A hierarquia usada nos *hubs* melhora a limitação de extensão de uma rede 10Base-T (100 m).

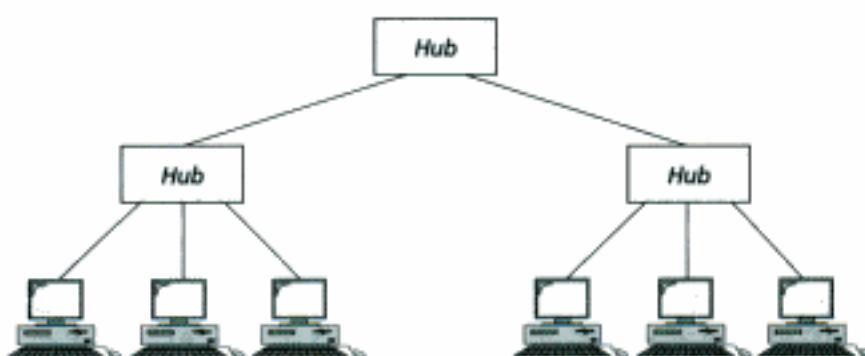


Figura 16.4 Hubs.

Bridges

Uma **bridge** opera tanto na camada física quanto na camada de enlace. Como um dispositivo da camada física, possui capacidade de regenerar na saída o sinal recebido na entrada. Além disso, como um dispositivo da camada de enlace a *bridge* verifica ("olha") o endereço físico (MAC) da origem e do destino contido no *frame*.

Filtragem

Qual é a diferença funcional entre uma *bridge* e um repetidor? Uma *bridge* possui capacidade de **filtragem**. Ela verifica os endereços de origem e de destino do *frame* e, baseada neles, toma decisões de encaminhamento dos *frames*. Toda vez que uma *bridge* recebe um *frame* ela "sabe" para qual porta deve encaminhá-lo, evitando inundar as outras portas com um *frame* que não se destina a elas. Uma *bridge* toma decisões de encaminhamento baseada numa tabela que associa endereços físicos às portas da *bridge*.

As bridges possuem tabelas utilizadas na tomada de decisões de encaminhamento de frames.

Vamos exemplificar a questão da tabela. Na Figura 16.5, duas LANs são interligadas através de uma *bridge*.

Se um *frame* destinado à estação 712B1345642 chega à porta 1, a *bridge* consulta sua tabela interna para determinar a porta de saída do *frame*. De acordo com a tabela ilustrada no desenho, os *frames* para 712B1345642 devem sair da *bridge* pela porta 1. Assim, o *frame* é encaminhado direto para a mesma porta por onde veio. De outro modo, se um *frame* para 712B1345642 chegar à porta 2, a porta de destino é a porta 1 e o *frame* é encaminhado para lá. No primeiro caso, a LAN2 permanece livre de tráfego. No segundo, ambos segmentos de LAN têm tráfego. No exemplo, ilustramos uma *bridge* de duas portas. Na maioria dos casos reais uma *bridge* possui mais de duas portas.

Uma bridge não modifica o endereço físico (MAC) inscrito nos frames.

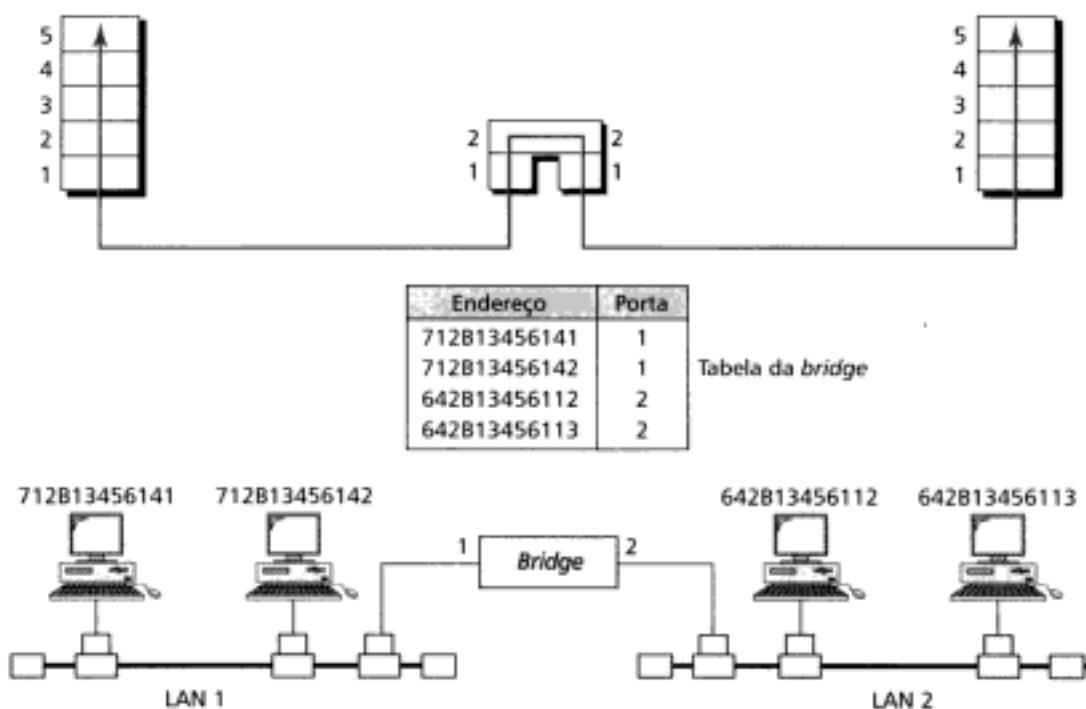


FIGURA 16.5 Bridge.

Bridges Transparentes

Uma **bridge transparente** é aquela na qual as estações não tomam conhecimento da existência da *bridge*. Se uma *bridge* é adicionada ou retirada do sistema não é necessário reconfigurar as estações. De acordo com a especificação IEEE 802.1d, um sistema equipado com *bridges* transparentes deve obedecer a três critérios:

- 1 Os *frames* devem ser encaminhados de uma estação à outra.
- 2 A tabela de encaminhamento deve ser aprendida e atualizada pela *bridge* após cada movimentação de *frame* na rede.
- 3 O sistema deve estar prevenido contra *loops*.

Encaminhamento (Forwarding) Uma *bridge* transparente deve encaminhar os *frames* conforme discutido na seção anterior.

Aprendizagem (Learning) As primeiras *bridges* colocadas no mercado tinham tabelas de encaminhamento estáticas. Os administradores de rede entravam manualmente com a tabela durante a operação de configuração (*setup*) das mesmas. Embora o processo de configuração fosse simples, não era prático atualizar manualmente as tabelas das *bridges*. Se uma estação era acrescentada ou retirada do sistema, a tabela tinha que ser modificada manualmente. O mesmo acontecia se uma estação tivesse o endereço MAC modificado, o que não é um evento raro. Por exemplo, durante a substituição do adaptador de rede, o novo adaptador possui um endereço MAC totalmente diferente do anterior.

Uma solução melhor do que a tabela estática é utilizar uma tabela dinâmica que endereça as portas automaticamente. Para tornar a tabela dinâmica é necessário que uma *bridge* aprenda os endereços gradualmente a partir da movimentação de *frames* através dela. Para tanto, a *bridge* inspeciona tanto o endereço de origem quanto o endereço de destino. O endereço de destino é utilizado na tomada de decisão de encaminhamento, enquanto que o endereço de origem é utilizado para adicionar entradas na tabela e para propósitos de atualização. Vamos elaborar o processo de construção e atualização da tabela baseados na Figura 16.6.

1. Quando a estação A transmite um *frame* para a estação D, a *bridge* desconhece todos os endereços, pois não dispõe de tabela de encaminhamento. Nesse caso, o *frame* é encaminhado para todas as três portas, inundando a rede. Contudo, olhando o endereço de origem, a *bridge* aprende que a estação A está conectada na porta 1. Significa que, no futuro, os *frames* destinados à estação A devem ser enviados através da porta 1. Assim, a *bridge* adiciona esta entrada na tabela. Tecnicamente, dizemos que a tabela recebeu a primeira entrada dinâmica.
2. Quando a estação E transmite um *frame* para a estação A, a *bridge* possui a entrada para a estação A. Então, o *frame* é encaminhado diretamente à porta 1. Nesse caso, não houve inundação da rede. Além disso, a *bridge* toma o endereço de origem da estação E e adiciona a segunda entrada dinâmica na tabela.
3. Quando a estação B transmite um *frame* para a estação C, a *bridge* não dispõe da entrada para a estação C. Outra vez, a rede é inundada de *frames* e a entrada dinâmica referente à estação B é acrescentada na tabela.
4. O processo de aprendizagem (*learning*) continua até que a *bridge* aprenda todos os endereços da rede para encaminhamento de *frames*.

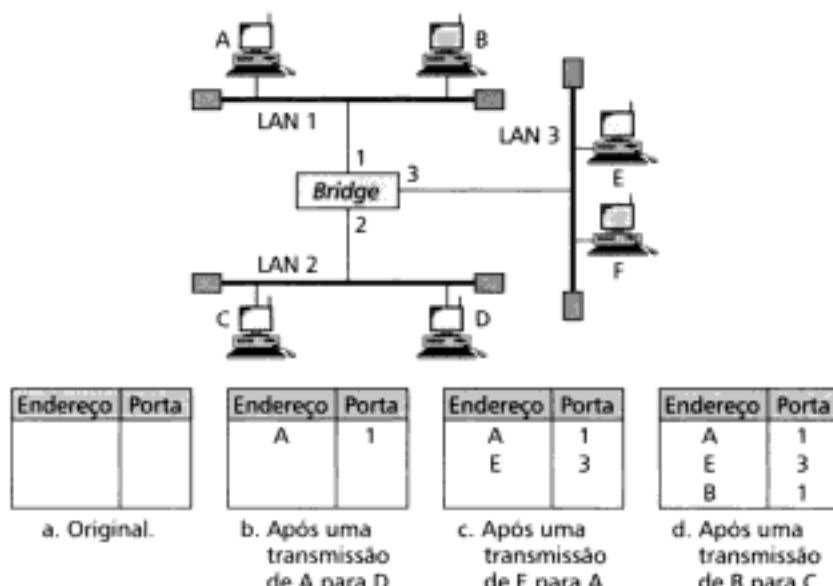


Figura 16.6 Aprendizagem da bridge.

Problema do Loop Bridges transparentes funcionam muito bem desde que não existam bridges redundantes no sistema. É uma tarefa comum aos administradores de rede utilizar bridges redundantes (mais de uma bridge entre um par de LANs) de modo a tornar o sistema mais confiável. Se, por alguma razão, uma bridge cair, a outra toma o seu lugar até que a falha seja reparada. Contudo, a redundância gera loops no sistema, o que é bastante indesejável. A Figura 16.7 ilustra um exemplo bem simples de como acontece loop entre duas LANs conectadas por duas bridges.

1. A estação A transmite um frame para a estação D. As tabelas de ambas bridges estão vazias. Ambas encaminham o frame e atualizam as respectivas tabelas baseadas no endereço de origem A.
2. Nesse caso, existem dois frames idênticos na LAN2. Sendo assim, a cópia enviada pela bridge 1 é recebida pela bridge 2, a qual não possui informação sobre o endereço de destino da estação D. Logo, a bridge 2 é inundada e retransmite o frame à LAN1. A cópia transmitida pela bridge 2 é recebida pela bridge 1 e é transmitida sem que a informação sobre a estação D seja atualizada na tabela de encaminhamento da bridge 1. A bridge 1 é inundada e também retransmite o frame à LAN1. Perceba que cada frame é transmitido separadamente visto que as bridges, como dois nós numa rede cujo meio é compartilhado, utilizam um método de acesso tal como o CSMA/CD. As tabelas de ambas bridges são atualizadas, mas ainda não há nenhuma informação sobre a estação de destino D.
3. Nesse ponto é a LAN1 que possui frames idênticos. O passo número 2 é repetido e ambas cópias inundam a rede outra vez, novamente na direção da LAN2.
4. O processo permanece continuamente. Já que as bridges também têm incorporadas o recurso dos repetidores, os frames são regenerados a cada iteração.

Para resolver o problema do loop, a especificação IEEE requer que bridges tenham residente o protocolo *spanning tree* para criar topologias imunes aos loops.

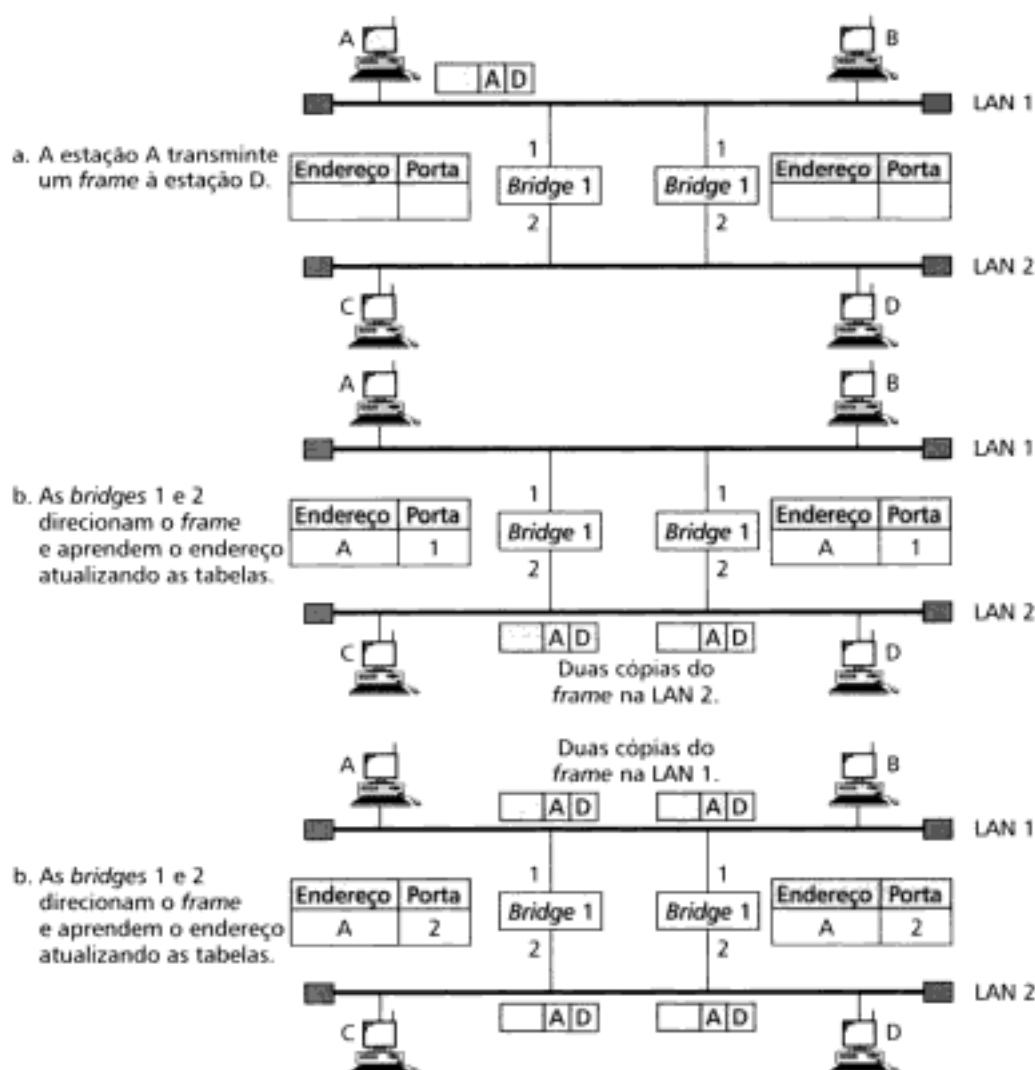


Figura 16.7 Problema do loop.

Algoritmo Spanning Tree

Na teoria dos grafos, o **spanning tree** é um grafo sem *loops*. Numa LAN, controlada por *bridge*, usar o *spanning tree* significa criar uma topologia onde cada LAN é acessada pelas outras LANs através de um único caminho, isto é, sem *loops*. Na maioria das vezes, não podemos modificar a topologia física de uma rede, devido às conexões físicas entre cabos e *bridges*, mas podemos criar uma topologia lógica sobreposta hierarquicamente à topologia física. O processo envolve três etapas:

1. A cada *bridge* é associado um identificador ID. A *bridge* com o menor ID é escolhida ou eleita como a *bridge raiz* (*root bridge*), assim como a raiz de uma árvore.
2. Identificamos uma porta de cada *bridge* (exceto da *bridge raiz*) como uma *porta raiz* (*root port*). A porta raiz é aquela com o menor custo no caminho da *bridge* até a *bridge* raiz. A determinação da porta com menor custo é feita pelo administrador de rede ou do sistema que, geralmente, leva em consideração a largura de banda do *link* em questão, o número mínimo de saltos da *bridge* até a LAN e/ou o caminho com o menor atraso de propagação (*delay*) de frames. Se duas portas tiverem o mesmo custo, o administrador simplesmente escolhe uma das portas como raiz.

3. Escolha uma *bridge designada* para cada LAN. Uma *bridge designada* é aquela que possui o menor custo entre a LAN e a *bridge raiz*. Nomeie a porta correspondente como *porta designada* (ou seja, a porta que conecta a LAN à *bridge designada*). Se duas *bridges* tiverem o mesmo custo, escolha aquela com menor ID.
4. Marque a porta raiz e a porta designada como *portas de encaminhamento (forwarding ports)*. As outras portas são as *portas de bloqueio (blocking ports)*. Uma ***forwarding port*** encaminha o *frame* que ela recebeu, já a ***blocking port*** não.

Vamos exemplificar a situação. O algoritmo escrito em linguagem C pode ser encontrado em Gilberg and Forouzan, *Data Structures: With Pseudocode Using C*, Thomson Learning. Na Figura 16.8 temos quatro LANs e cinco *bridges*.

A Figura 16.9 ilustra as três primeiras etapas. Elegemos a *bridge* B1 como a *bridge raiz* assumindo que ela tem o menor ID. As portas raiz são assinaladas com uma única estrela. As *bridges* designadas têm uma seta apontando para elas a partir da LAN correspondente. Por último, as portas designadas são marcadas por duas estrelas.

Assim, podemos identificar as portas raiz e portas designadas como portas de encaminhamento de *frames (forwarding ports)*. As demais serão todas portas de bloqueio (*blocking ports*). Na Figura 16.10 mostramos uma porta de bloqueio como uma linha tracejada. A conexão física continua existindo, mas a *bridge* nunca encaminha *frames* através dessas portas.

Você pode verificar que existe um único caminho entre quaisquer pares de LANs no sistema utilizando o algoritmo *spanning tree*. Desse modo, fica impossível a existência de *loops* já que através de um único caminho, isto é, sem realimentação, não podemos formar retornos (*loops*). Para verificar o que dissemos acima, comece olhando todas as possibilidades da LAN1 até a LAN2, depois da LAN1 até a LAN3 ou LAN4 e assim por diante.

Algoritmo Dinâmico Nossa descrição do algoritmo *spanning tree* pode ter sugerido que o sistema requer entradas manuais para o correto funcionamento. Isto não é verdade. Cada *bridge* possui um *software* residente de gerenciamento automático do processo: o protocolo *spanning tree*. As

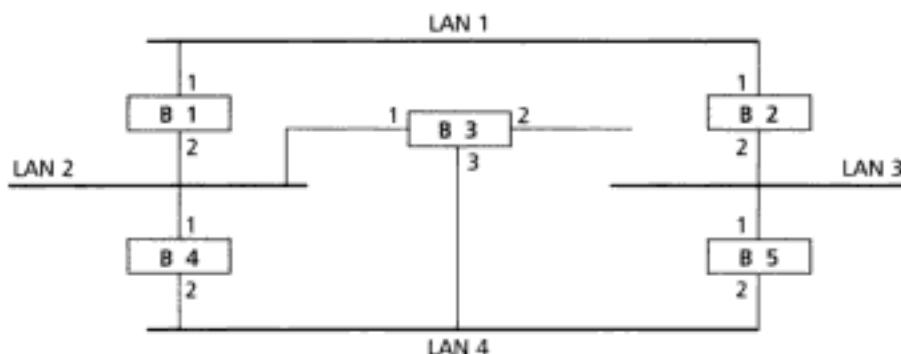


FIGURA 16.8 Aplicação do protocolo *spanning tree*.

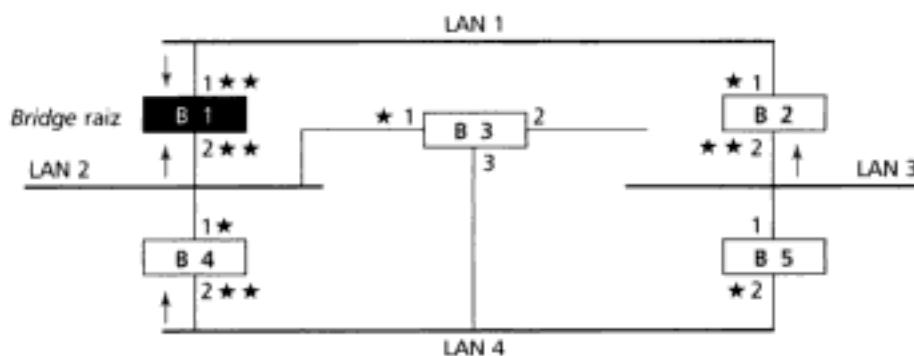


FIGURA 16.9 Aplicando o protocolo *spanning tree*.

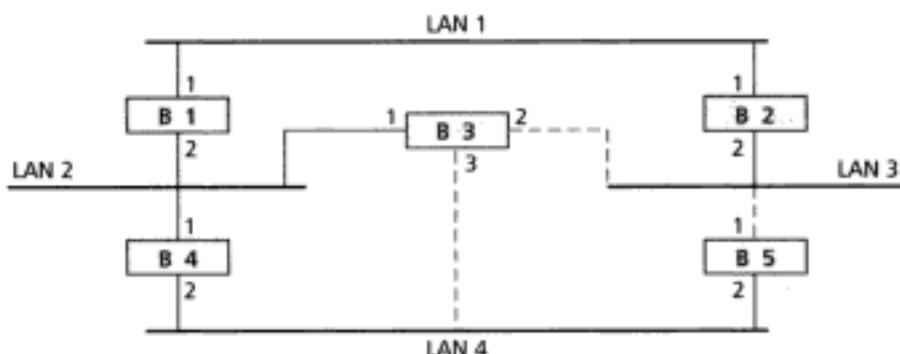


FIGURA 16.10 Modos de operação das bridges: *forwarding* e *blocking*.

bridges trocam informações através do que denominamos Bridge Protocol Data Unit (BPDUs) para atualizar o mecanismo de *spanning tree*. O *spanning tree* é atualizado toda vez que ocorre uma mudança no sistema, tal como uma falha, adição ou remoção de uma bridge do sistema.

Source Routing Bridge – SRB

Outro modo de inibir loops em redes conectadas com redundância de bridges é utilizar o **Source Routing Bridge (SRB)**. Dentre as tarefas de uma bridge transparente estão incluídos a filtragem, encaminhamento e bloqueio de frames. Num sistema utilizando SRB, tais tarefas são realizadas pela estação de origem.

No SRB, uma estação transmissora define as bridges que o frame deve visitar. Os endereços dessas bridges são anexados aos frames. Noutras palavras, o frame contém não somente os endereços de origem e de destino, mas também os endereços de todas as bridges que ele deve visitar.

A estação de origem obtém os endereços das bridges através de trocas de frames especiais cujas prioridades são maiores que as prioridades dos frames de dados.

O esquema SRB foi padronizado pelo IEEE para ser utilizado nas LANs Token Ring. Estas LANs não são muito comuns hoje em dia.

Bridges Interconectando Diferentes LANs

Teoricamente, uma bridge deveria ser capaz de conectar LANs de diferentes protocolos de camada de enlace, tal como uma LAN Ethernet a uma LAN sem fio. Entretanto, devem ser consideradas muitas questões:

- **Formato do frame.** Cada tipo de LAN tem um padrão de frame próprio (compare um frame Ethernet com um frame WLAN).
- **Tamanho máximo do frame de dados.** Se o tamanho de um frame recebido de uma LAN for grande demais para a LAN de destino, há necessidade de fragmentação de dados. Assim, também há necessidade de remontagem de dados na estação de destino. Veremos no Capítulo 19 que isto só é permitido na camada de rede. Então, uma bridge deve descartar todos os frames considerados grandes demais para o sistema a qual ela pertence.
- **Taxa de transmissão de dados.** Cada tipo de LAN tem uma taxa de transmissão própria (compare a taxa de transmissão da LAN Ethernet padrão, isto é, 10 Mbps, com a taxa típica de uma WLAN, 1 Mbps). Uma bridge deve dispor de buffer suficiente para compensar esta diferença.
- **Ordem dos bits.** Cada tipo de LAN possui uma estratégia de transmissão de bits. Em algumas LANs, os bits mais significativos são transmitidos no primeiro byte. Noutras LANs, os bits menos significativos são transmitidos no primeiro byte.

- **Segurança.** Certas LANs, tal como as WLANs, possuem medidas de segurança implementadas na camada de enlace. Outras LANs não, como a Ethernet. Segurança geralmente envolve algum nível de criptografia (veja Capítulo 29). Quando uma *bridge* recebe um *frame* oriundo de uma WLAN, ela deve descriptografar a mensagem antes de encaminhá-lo à rede Ethernet.
- **Suporte à multimídia.** Algumas LANs suportam recursos de multimídia e qualidade de serviços (QoS) necessários a este tipo de comunicação. Outras não.

Switch de Camada 2

Quando utilizamos o termo *switch* devemos tomar certos cuidados porque *switch* pode significar dois dispositivos diferentes. Vamos esclarecer o termo adicionando ao nome do dispositivo a camada onde o *switch* opera. Os *switches* podem ser dispositivos de camada 2 ou 3. Um **switch de camada 3** é um dispositivo usado na camada de rede. O **switch de camada 2** implementa as camadas física e de enlace.

Um *switch* de camada 2 nada mais é do que uma *bridge* multiportas projetada para melhorar a *performance* de uma rede. Uma *bridge* com poucas portas pode conectar pequenas LANs. Uma *bridge* multiportas (*switch* de camada 2) é capaz de alocar uma estação em cada porta, e cada estação sendo uma entidade independente. Assim, não há competição entre estações para ver quem vai utilizar o meio de transmissão (diríamos sem colisão para as redes Ethernet que utilizam o método CSMA/CD). Neste livro, para evitar quaisquer tipos de confusão utilizaremos o termo *bridge* quando estivermos fazendo menção ao *switch* de camada 2.

Mais informações sobre *switches* são fornecidas no Capítulo 19, quando estivermos estudando roteadores, e no Apêndice F.

Roteador e Switches de Camada 3

Nossa discussão sobre **roteadores** e **switches de camada 3** será adiada até cobrirmos a camada de rede no Capítulo 19, 21 e Apêndice F.

16.2 REDES BACKBONES

Alguns dos dispositivos discutidos neste capítulo podem ser utilizados para conectar LANs de modo a formar uma rede *backbone* ou simplesmente um *backbone**. Um *backbone* permite que várias LANs sejam conectadas juntas. A regra essencial é não conectar estações diretamente ao *backbone*. Assim, as estações serão parte de uma LAN e o *backbone* é utilizado para conectar as LANs. O próprio *backbone* pode ser considerado uma LAN utilizando algum padrão de rede local, como o Ethernet, e cada conexão ao *backbone* é uma LAN.

Embora existam muitas arquiteturas diferentes de *backbones*, veremos somente as duas mais comuns: *backbone barramento* e *backbone estrela*.

Backbone Barramento

Em um **backbone barramento**, a topologia barramento é utilizada na estruturação do *backbone*. O *backbone* pode utilizar um dos padrões de redes que suportam topologia barramento, tal como 10Base2 e 10Base5.

* N. de R. T.: Um *backbone* é um tronco principal (espinha dorsal) que conecta os nós de uma rede.

Um *backbone* barramento utiliza a topologia barramento na estruturação do *backbone*.

Backbones são utilizados como *backbone* de distribuição para conectar diferentes construções. Cada uma dessas construções pode ser formada de uma única LAN ou então outro *backbone* (normalmente estrela). Um bom exemplo de *backbone* é aquele utilizado para interligar andares de prédios e unidades dentro de um campus universitário. Cada andar de um prédio tem pelo menos uma rede local e o prédio como um todo geralmente tem um *backbone* vertical interligando todos os andares. Um *backbone* barramento é capaz de interligar todas as LANs formadas nos andares de um prédio, assim como os outros *backbones* que chegam até os nós da rede. A Figura 16.11 ilustra um exemplo de *backbone* barramento conectando quatro LANs por meio de *bridges*.

Na figura anterior, se uma estação pertencente a uma LAN necessita enviar *frames* para outra estação localizada dentro da mesma LAN, a porta da *bridge* correspondente, que possibilita o acesso da LAN ao *backbone*, deve entrar em modo de bloqueio. Entretanto, se uma estação tiver *frames* para enviar a outra estação localizada numa LAN diferente, a *bridge* de acesso ao *backbone* tem que encaminhar o *frame* ao *backbone* para que chegue à *bridge* que leva à LAN de destino. A *bridge* conectando o *backbone* à LAN (onde está a estação destino), se encarrega de entregar o *frame*. Cada *bridge* conectada ao *backbone* possui uma tabela que mostra as estações no lado da LAN. O bloqueio ou encaminhamento dos *frames* está baseado no conteúdo dessa tabela.

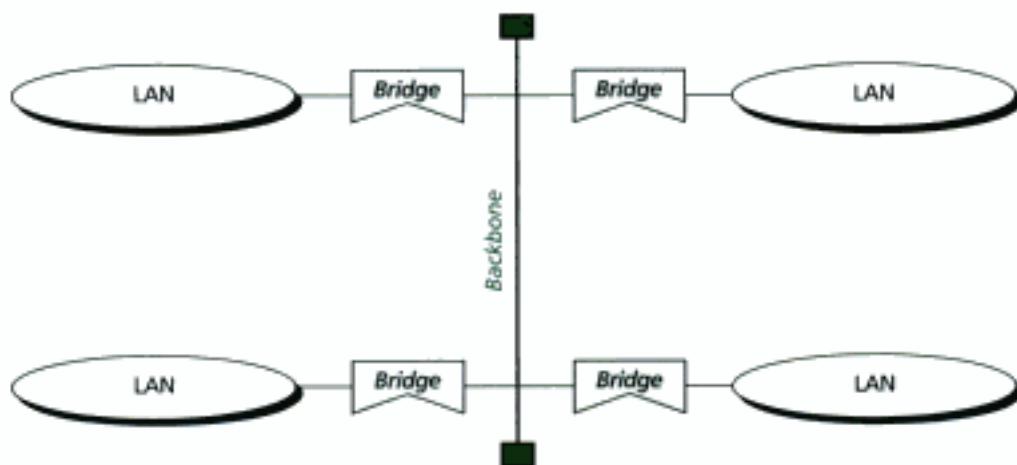


Figura 16.11 Backbone linear.

Backbone Estrela

Num **backbone estrela**, às vezes denominado *switched backbone*, é utilizada a topologia estrela. Nesta configuração, o *backbone* é realizado através de um *switch* conectando todas as LANs.

Em um *backbone* estrela, a topologia do *backbone* é estrela; o *backbone* é implementado através de um *switch*.

A Figura 16.12 ilustra um *backbone* estrela. Nesta configuração, perceba que o *switch*, além de implementar o *backbone*, permite a interligação de todas as LANs.

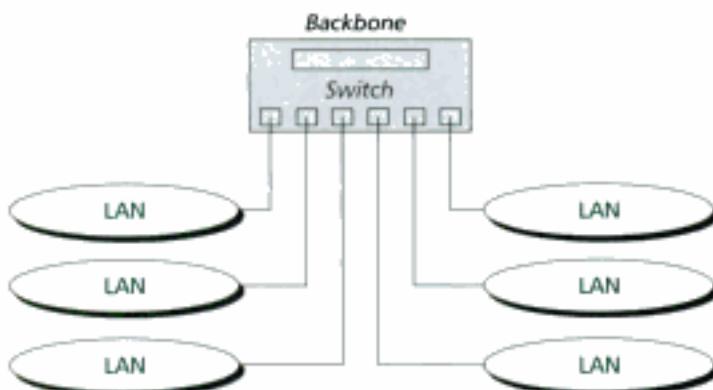


Figura 16.12 Backbone estrela.

Backbones estrela são utilizados com muita freqüência dentro de uma construção. Num prédio com muitos andares, por exemplo, encontramos normalmente uma LAN servindo a cada andar em particular. Nesse caso, o *backbone* (o *switch*) pode ser instalado na base do primeiro andar e receber cabos provenientes dos *switches* de todas as outras LANs localizadas em andares diferentes. Se as LANs têm individualmente topologia física estrela, os *switches* podem ser instalados no ponto de distribuição de cada andar. Freqüentemente, encontramos um *rack* ou *chassis* na base do prédio onde o *backbone* estrela e todos os *hubs* ou *switches* são instalados.

Interligando LANs Remotas

Outra aplicação bastante comum para os *backbones* acontece durante a interligação LANs remotas. Por exemplo, este tipo de *backbone* é bastante útil quando uma empresa tem muitos setores contendo LANs espalhadas numa determinada região e ela deseja conectá-los. Essa conexão muitas vezes se dá através de *bridges*, às vezes denominadas **bridges remotas**. Essas *bridges* agem como dispositivos conectando LANs e redes ponto a ponto, tais como linhas telefônicas dedicadas e linhas ADSL. A rede ponto a ponto, nesse caso, pode ser considerada como um LAN sem estações. A Figura 6.13 ilustra um *backbone* conectando LANs remotas.

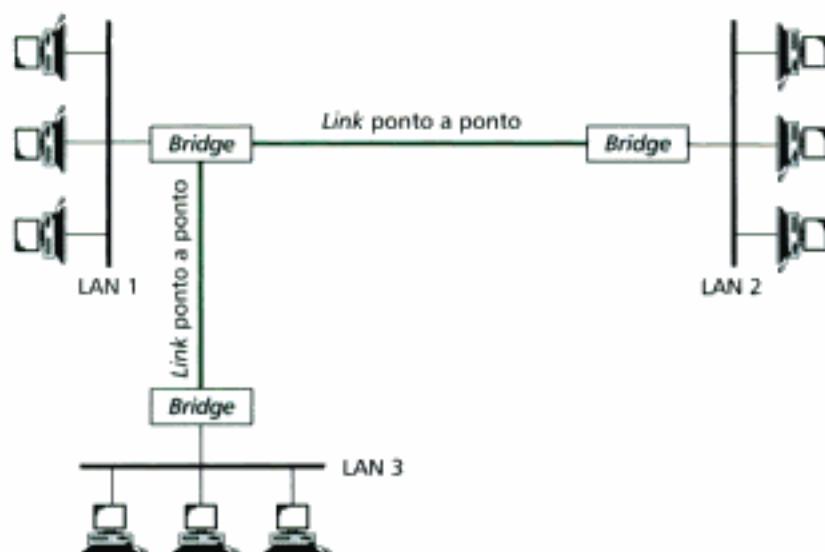


Figura 16.13 Interligando LANs remotas.

Uma rede ponto a ponto funciona como uma LAN sem estações conectada através de *bridges* remotas.

16.3 LANS VIRTUAIS (VLANS)

Uma estação é considerada parte de uma LAN se ela pertence fisicamente a esta LAN. Este critério de associação é puramente geográfico. O que acontece se necessitarmos de uma conexão virtual entre duas estações pertencentes a LANs fisicamente diferentes? Utilizaremos a questão anterior para definir o conceito de **Rede Local Virtual (VLAN)** como uma LAN configurada logicamente via *software*, não através de fios.

Vamos utilizar um exemplo para refinar esta definição. A Figura 16.14 mostra uma LAN conectada por *switch* numa empresa de engenharia onde 10 estações foram agrupadas em três LANs menores conectadas ao *switch*. As quatro primeiras estações trabalham juntas no primeiro grupo. O mesmo acontece com as três estações das outras duas LANs: elas formam mais dois grupos diferentes contendo três estações cada. A LAN foi configurada de modo a permitir este arranjo físico.

Entretanto, o que acontece se o administrador da rede tiver que mover dois engenheiros do primeiro para o terceiro grupo para aumentar o desempenho das atividades desenvolvidas nesse grupo? Assumindo a solução de praxe, a configuração física da LAN deveria ser modificada. O técnico de suporte da rede deveria lançar o cabeamento novamente. Numa LAN tradicional, mudanças nos grupos de trabalho refletem-se como mudanças na configuração da rede.

A Figura 16.15 mostra a mesma LAN conectada por *switch* e dividida em VLANs. A idéia central da tecnologia das VLANs é dividir uma LAN de maiores proporções em segmentos lógicos, ao invés de segmentos físicos. Uma LAN pode ser dividida em muitas LANs lógicas denominadas VLANs. Cada VLAN pode formar um grupo de trabalho dentro de uma empresa. Se um usuário tiver que ser movido de um grupo para outro, não há necessidade de lançar novo cabeamento físico, basta reconfigurar, via *software*, o novo grupo de trabalho desse usuário. Qualquer estação pode ser movida logicamente para outra VLAN. Todos os membros de uma VLAN podem receber mensagens de *broadcast* enviadas nessa VLAN particular. Isto significa que se uma estação for desligada da VLAN1 e associada à VLAN2, ela passará a receber mensagens de *broadcast* enviadas para a VLAN2 e não da VLAN1 (a original).

No exemplo da empresa de engenharia, a solução através da utilização de VLANs é a mais fácil e prática. Basta mover, via *software*, os engenheiros que trabalham nas duas estações do grupo 1 para o grupo 3. Fisicamente, as estações desses engenheiros continuarão nos mesmos lugares que sempre estiveram e nenhum cabeamento adicional foi lançado para a nova configuração.

A tecnologia de VLANs permite até mesmo agrupar, numa VLAN, estações conectadas em diferentes *switches*. A Figura 16.16 ilustra um *backbone* local interligando dois *switches* e três

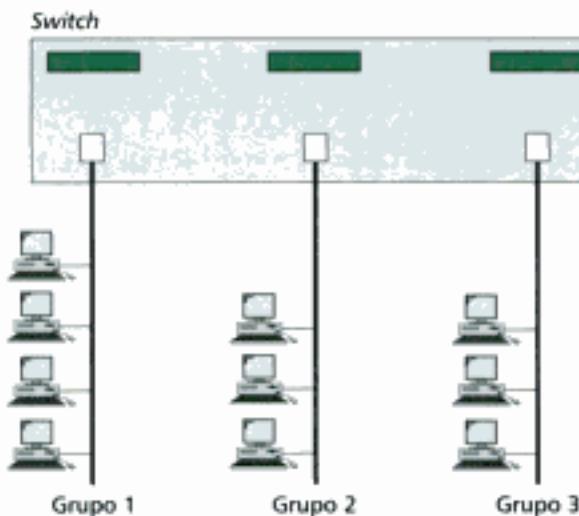


FIGURA 16.14 Conexão de três LANs através de um *switch*.

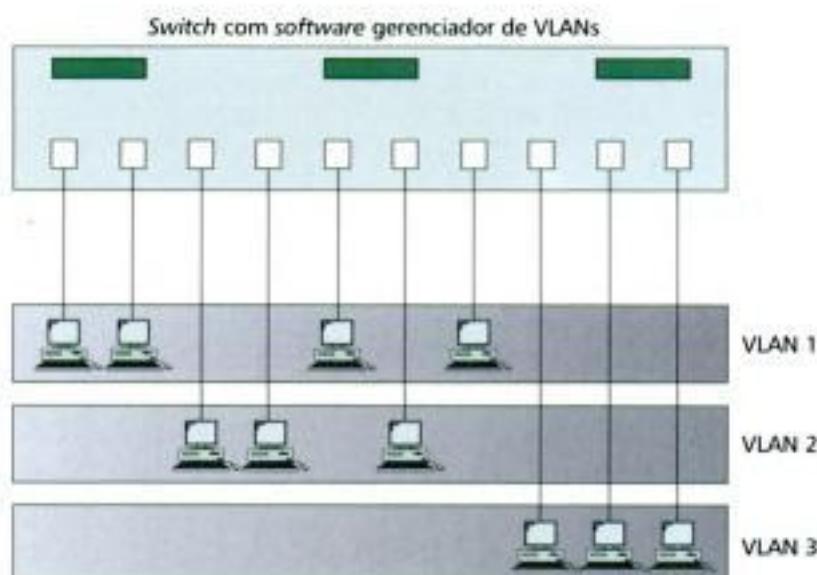


FIGURA 16.15 Switch com software gerenciador de VLANs.

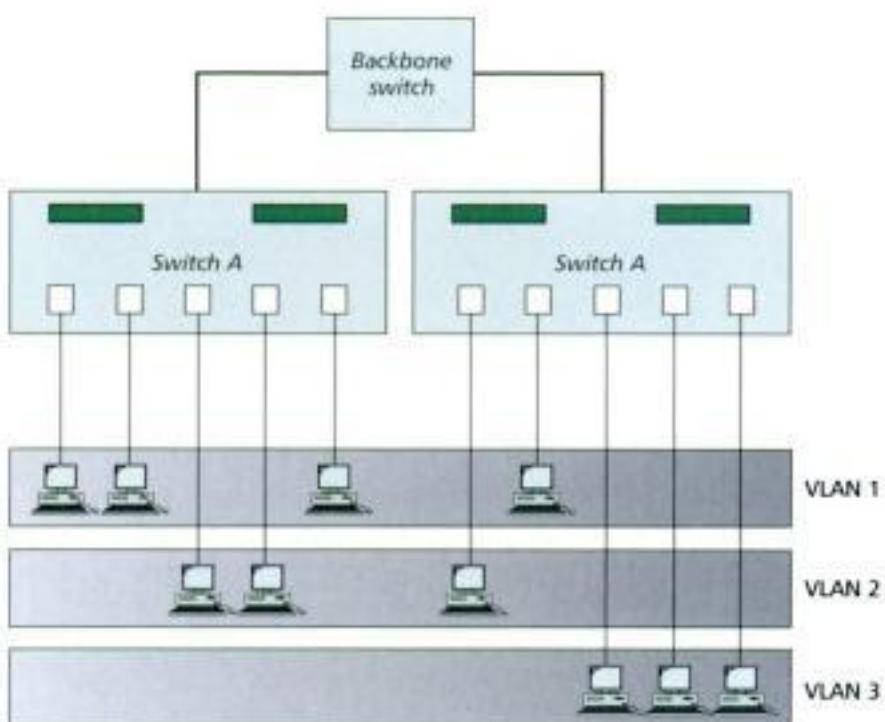


FIGURA 16.16 Dois switches num backbone usando software gerenciador de VLANs.

VLANs. As estações dos *switches* A e B podem pertencer a qualquer uma das VLANs implementadas no esquema.

Aliás, esta é uma excelente solução para uma empresa que ocupa fisicamente duas construções separadas. Cada uma dessas construções possui uma rede LAN estruturada através de um *switch*, que são conectadas pelo *backbone*. Usuários diferentes nas duas partes da empresa podem pertencer ao mesmo grupo de trabalho até mesmo quando eles estiverem em LANs fisicamente diferentes.

Partindo dos três exemplos, podemos definir uma característica típica de VLANs:

VLANs reduzem domínios de broadcast.

As VLANs agrupam estações pertencentes a uma ou mais LAN física para formar domínios de *broadcast*. As estações numa VLAN se comunicam umas com as outras, embora, muitas vezes, elas pertençam a segmentos físicos diferentes.

Agrupamento Lógico de Usuários e Recursos

Que características podem ser utilizadas para agrupar estações durante a criação das VLAN? Os fabricantes utilizam muitos parâmetros diferentes, tais como: números das portas, endereços MAC, endereços IP, endereço IP *multicast* ou combinações dessas.

Números das Portas

Alguns fabricantes estabelecem os números das portas do(s) *switch*(es) como parâmetro de agrupamento lógico. Por exemplo, um administrador de rede pode definir que as estações conectadas nas portas 1, 2, 3 e 7 pertencem à VLAN1; as estações conectadas nas portas 4, 10 e 12 pertencem à VLAN2 e assim por diante.

Endereços MAC

Também é bastante comum encontrar *switches* que utilizam permitem o agrupamento através do endereço físico de 48-bits (endereço MAC). Por exemplo, um administrador de rede pode definir que as estações cujos endereços MAC são E21342A12334 e F2A123BCD341 pertencem à VLAN1.

Endereços IP

Alguns fabricantes de *switches* utilizam o endereço IP de 32-bits (veja Capítulo 19) como parâmetro de agrupamento lógico. Por exemplo, um administrador pode definir que as estações cujos endereços IP são 181.34.23.67, 181.34.23.98 e 181.34.23.112 pertencem à VLAN1.

Endereços IP Multicast

Alguns fabricantes de *switches* utilizam o endereço IP *multicast* (veja Capítulo 19) como parâmetro de agrupamento lógico. Nesse caso, o *multicasting* da camada de rede é traduzido em *multicast* na camada de enlace.

Combinação

Hoje em dia é muito comum encontrarmos equipamentos que permitem utilizar todos esses parâmetros via *software* residente nos *switches*. O administrador conecta-se através da porta de consóleo e configura os parâmetros que deseja utilizar no agrupamento.

Configuração

Como configurar as estações para formarem diferentes VLANs? As estações podem ser configuradas de três maneiras distintas: manual, semi-automática e automática.

Configuração Manual

Num procedimento de configuração manual, o administrador de rede utiliza o *software* de VLAN para configurar manualmente as estações para a VLAN especificada no projeto lógico da rede. Qualquer migração posterior de uma VLAN para outra é feita manualmente. Note que isso não é uma configuração física da rede! O termo *manualmente* sugere que o administrador deve digitar os números das portas, os endereços IP ou outros parâmetros usando o *software* de criação das VLANs.

Configuração Automática

Numa configuração automática, as estações são conectadas ou desconectadas automaticamente de uma VLAN usando algum critério preestabelecido pelo administrador de rede. Por exemplo, um administrador de rede pode definir o número de um projeto como critério de associação a um grupo. Quando um usuário mudar de projeto, ele ou ela migra automaticamente para uma nova VLAN.

Configuração Semi-Automática

A configuração semi-automática reúne características das outras duas configurações: manual e automática. Usualmente, o procedimento inicial é todo manual e, posteriormente, as migrações das estações são feitas de modo automático.

Identificação de VLANs

Numa rede totalmente conectada por *switches*, cada *switch* deve conhecer não somente as estações associadas às VLANs que ele gerencia, mas também as outras estações nas VLANs gerenciadas pelos outros *switches*. Na Figura 16.16, por exemplo, o *switch* A deve reter informações sobre o *status* das estações associadas às VLANs do *switch* B e vice-versa. Existem três métodos que possibilitam essa comunicação: manutenção da tabela, identificação de *frames* (*frame tagging*) e Multiplexação por Divisão do Tempo (TDM).

Manutenção da Tabela

Neste método, quando uma estação envia um *frame* de broadcast para os membros do grupo a qual ela pertence, o *switch* cria uma entrada numa tabela e grava o endereço da estação. Os *switches* trocam essas tabelas periodicamente entre eles, mantendo-as atualizadas.

Frame Tagging

Este método baseia-se na adição de informação ao cabeçalho dos *frames* numa rede. Quando um *frame* viaja entre os *switches*, o cabeçalho extra é adicionado ao MAC *frame* para definir a VLAN de destino. A informação contida no *frame* de identificação (*frame tagging*) é utilizada pelos *switches* para determinar as VLANs que devem receber a mensagem de broadcast.

Multiplexação por Divisão do Tempo (TDM)

Neste método, uma conexão entre *switches*, denominada *trunk*, é dividida em canais e compartilhada no tempo (veja TDM no Capítulo 6). Por exemplo, suponha que um sistema é formado de cinco VLANs conectadas num *backbone*. Sendo assim, nesse caso, cada *trunk* é dividido em cinco canais. O tráfego destinado à VLAN1 viaja através do canal 1, o tráfego destino à VLAN2 viaja no canal 2 e assim por diante. No *trunk*, o *switch* receptor determina a VLAN de destino verificando o canal no qual o *frame* foi recebido.

Padrão IEEE 802.1Q

Em 1996, o subcomitê IEEE 802.1 especificou um padrão, denominado 802.1Q, que define o formato do *frame tagging*. O padrão também define o formato a ser utilizado nos *backbones multi-switches* e regulamenta o uso de equipamentos de fabricantes diferentes. O padrão IEEE 802.1Q promoveu a padronização de outras questões relacionadas às VLANs. Muitos fabricantes aceitaram esse padrão.

Vantagens

Existem muitas vantagens relacionadas ao uso das VLANs, dentre elas citamos: redução de custos e de tempo, criação de grupos de trabalho virtuais e segurança.

Redução de Custos e Tempo

As VLANs podem reduzir o custo das migrações de estações entre grupos de uma rede. A reconfiguração física consome tempo e dinheiro. Ao invés de movimentar fisicamente estações para outro segmento de rede ou, até mesmo para outro *switch*, é muito mais fácil e rápido reconfigurá-las logicamente.

Criação de Grupos de Trabalho Virtuais

As VLANs permitem a criação de grupos de trabalho virtuais. Por exemplo, num ambiente universitário, professores trabalhando no mesmo projeto podem enviar mensagens de *broadcast* entre eles sem a necessidade de pertencerem a um mesmo departamento. Isto pode reduzir o tráfego se a capacidade de *multicasting* citada anteriormente estiver sendo utilizada.

Segurança

Talvez o maior benefício proporcionado pelas VLANs é o aumento do nível de segurança de uma rede. Usuários pertencentes ao mesmo grupo podem enviar mensagens de *broadcast* com a garantia de que os usuários nos outros grupos não as estarão recebendo.

16.4 TERMOS-CHAVE

Amplificador	Porta de encaminhamento (<i>forwarding port</i>)
Ativos de rede	Rede Local Virtual (<i>Virtual Local Area Network</i> – VLAN)
<i>Backbone</i> barramento	Repetidor
<i>Backbone</i> estrela	Roteador
<i>Bridge</i>	Segmento
<i>Bridge</i> remota	Source Routing Bridge (SRB)
<i>Bridge</i> transparente	<i>Spanning tree</i>
Filtragem	<i>Switch</i> de camada 2
<i>Hub</i>	<i>Switch</i> de camada 3
Porta de bloqueio (<i>blocking port</i>)	

16.5 RESUMO

- Um repetidor é o ativo de rede que opera na camada física da arquitetura TCP/IP. O repetidor regenera um sinal, conecta segmentos de uma LAN e não possui a capacidade de filtrar *frames*.
- Uma *bridge* é o ativo de rede que opera nas duas primeiras camadas do modelo TCP/IP (física e enlace).
- Uma *bridge* transparente tem a capacidade de encaminhar e filtrar *frames*, além de montar uma tabela de endereços dos dispositivos da rede.
- Uma *bridge* pode usar o algoritmo de *spanning tree* para criar topologias sem *loops*.
- Um *backbone* permite a interligação de muitas LANs.
- Um *backbone* usualmente possui topologia barramento ou estrela.
- Uma rede local virtual (VLAN) é configurada por *software*, não através de cabeamento físico.
- A associação de uma estação ou um recurso a um grupo da VLAN pode ser baseada nos números de portas, endereços MAC, endereços IP, endereços IP *multicast* ou combinações destes.
- As VLANs reduzem eficientemente os custos e tempos de implementação de uma rede. Além disso, podem reduzir o tráfego e proporcionar um nível a mais de segurança de rede.

16.6 PRATIQUE OS CONHECIMENTOS ADQUIRIDOS

Questões de Revisão

1. Qual é a diferença entre um repetidor e um amplificador?
2. O que queremos dizer quando afirmamos que uma *bridge* tem a capacidade de filtrar o tráfego? Qual é a importância da filtragem?
3. O que é uma *bridge* transparente?
4. De que maneira um repetidor pode estender o comprimento de uma LAN?
5. Como um *hub* está relacionado a um repetidor?
6. Qual é a diferença entre uma *bridge* raiz (*root bridge*) e uma *bridge* designada (*designated bridge*)?
7. Qual é a diferença entre uma porta de encaminhamento (*forwarding port*) e porta de bloqueio (*blocking port*)?
8. Qual é a diferença entre um *backbone* barramento e um *backbone* estrela?
9. Como uma VLAN pode reduzir tempo e dinheiro numa empresa?
10. Como uma VLAN proporciona segurança extra para uma rede?
11. Como uma VLAN reduz o tráfego de um rede?
12. Qual é a base utilizada para associação a uma VLAN?
13. De que forma o TDM está envolvido numa comunicação entre VLANs?

Questões de Múltipla Escolha

14. Das opções abaixo, qual melhor representa um dispositivo ativo de rede?
 - Bridge*
 - Repetidor
 - Hub*
 - Todas as opções anteriores
15. Uma *bridge* encaminha ou filtra um *frame* comparando a informação na tabela de endereços dela com _____ do *frame*.
 - Endereço origem (camada 2)
 - Endereço do nó físico de origem
 - Endereço de destino (camada 2)
 - Endereço de destino (camada 3)
16. Uma *bridge* pode _____.
 - Filtrar um *frame*
 - Encaminhar um *frame*
 - Estender uma LAN
 - Todas as respostas anteriores
17. Repetidores operam na(s) camada(s) _____.
 - Física
 - De enlace
 - De rede
 - (a) e (b)
18. De fato, um(a) _____ é um repetidor multiportas.
 - Bridge*
 - Roteador
 - VLAN
19. As *bridges* operam na(s) camada(s) _____.
 - Física
 - De enlace
 - De rede
 - (a) e (b)
20. Um repetidor recebe um sinal enfraquecido e o _____.
 - Amplifica
 - Regenera
 - Remonta
 - Roteia novamente
21. Uma *bridge* trabalha com o endereço _____ de uma estação numa rede.
 - Físico (MAC)
 - De enlace
 - De acesso
 - Todas as respostas anteriores
22. Uma rede conectada por *bridges* redundantes pode vir a ter problemas com _____ de *frames* no sistema.
 - Loops*
 - Filtros
 - Spanning trees*
 - Todas as respostas anteriores
23. A *bridge* _____ será aquela com menor ID.
 - Raiz

- b. Designada
 - c. De encaminhamento (*forwarding*)
 - d. De bloqueio (*blocking*)
24. A *bridge* com o menor custo na LAN e a *bridge* raiz são denominadas *bridges* _____.
- a. Designadas
 - b. De encaminhamento (*forwarding*)
 - c. De bloqueio (*blocking*)
 - d. (a) e (b)
25. Uma *bridge* nunca encaminha *frames* através da porta _____.
- a. Raiz
 - b. Designada
 - c. De encaminhamento (*forwarding*)
 - d. De bloqueio (*blocking*)
26. Que tipo de *bridge* constrói e atualiza sua tabela de endereços a partir do tráfego de *frames* na rede?
- a. Simples
 - b. Transparente
 - c. (a) e (b)
- d. Nenhuma opção anterior
27. A tecnologia VLAN divide uma LAN em grupos _____.
- a. Físicos
 - b. Lógicos
 - c. Multiplexados
 - d. Enquadrados
28. Que parâmetro das estações pode ser utilizado para agrupar estações numa VLAN?
- a. Números de portas
 - b. Endereços MAC
 - c. Endereços IP
 - d. Todas acima
29. Numa VLAN, as estações são separadas em grupos através de _____.
- a. Métodos físicos
 - b. Métodos lógicos
 - c. Localização
 - d. *Switches*

Exercícios

30. Complete a tabela da Figura 16.6 após cada uma das estações ter enviado um pacote às demais.
31. Construa um sistema formado por 3 LANs e 4 *bridges*. As *bridges* (B1 a B4) conectam as LANs do seguinte modo:
- a. B1 conecta as LANs de número 1 e 2.
 - b. B2 conecta as LANs de número 1 e 3.
 - c. B3 conecta as LANs de número 2 e 3.
 - d. B4 conecta as LANs de número 1, 2 e 3.
- Eleja B1 como *bridge* raiz. Aplicando o algoritmo de *spanning tree*, identifique as portas de encaminhamento e de bloqueio das *bridges*.

Telefonia Celular e Redes de Satélites

Estudamos as redes locais sem fio no Capítulo 15. A tecnologia sem fio (*wireless*) também é utilizada na telefonia celular e nas redes de satélites. Neste capítulo, discutiremos os padrões desse tipo de rede, assim como os diversos tipos de métodos de acesso de canalização (veja Capítulo 13). Finalmente, faremos uma breve explanação das redes de satélites, uma tecnologia que estará disponível tanto para telefonia celular como para o acesso direto à Internet.

17.1 TELEFONIA CELULAR

A **telefonia celular** foi projetada para proporcionar serviços de comunicação entre duas unidades móveis, chamadas Estações Móveis (MS), ou entre uma unidade móvel e uma unidade fixa, denominada freqüentemente de Estação Rádio-Base (ERB ou BS). Um provedor de serviços deve ser capaz de localizar e rastrear uma chamada, alocar canais às estações móveis e comutar os canais entre ERBs quando uma ou mais estações móveis em conversação estiverem movendo-se para fora da célula que mantém a chamada.

Para tornar possível o rastreamento das MS, cada área de serviço celular é dividida em pequenas regiões denominadas células. Cada célula possui uma antena controlada por uma central, denominada **Central de Comutação e Controle (Mobile Switching Center – MSC)**. A MSC coordena a comunicação entre todas ERBs e a central telefônica responsável pela conexão das chamadas, pelo registro das informações e pela tarifação (veja Figura 17.1).

O tamanho da célula não é fixo e pode ser aumentado ou diminuído dependendo da população da área de cobertura. O raio típico de uma célula gira entre 2 e 15 km. Áreas cuja densidade populacional é maior requerem células menores, geograficamente distribuídas, de modo a suportar a demanda. Uma vez determinado, o tamanho de uma célula é otimizado para evitar interferência entre os sinais das células adjacentes. Para tanto, a potência de transmissão de cada torre é mantida em níveis baixos, mas suficiente para cobrir toda a célula.

Princípio de Reuso de Freqüências

Em geral, células vizinhas não podem operar no mesmo conjunto de freqüências para comunicação porque isso pode criar interferência nas chamadas dos usuários próximos à fronteira entre células. Entretanto, o conjunto de freqüências disponíveis é limitado e há necessidade de reutilizar

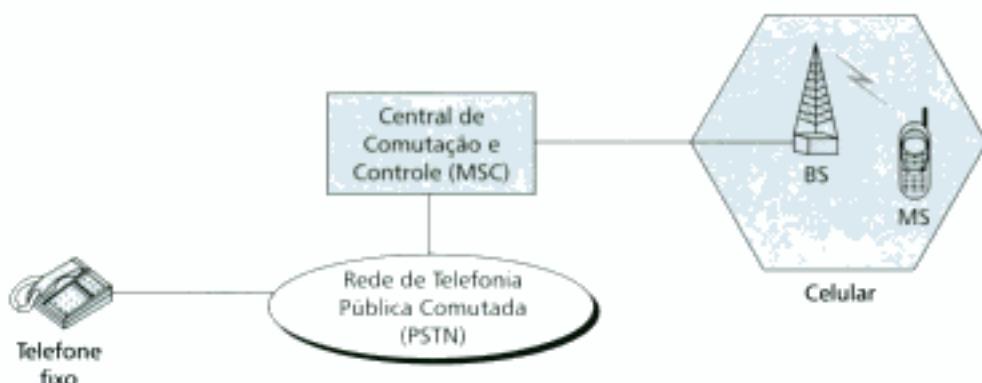


Figura 17.1 Sistema de telefonia celular.

as freqüências. Um padrão de freqüências de reuso é uma configuração de N células (um *cluster*), onde N representa o **fator de reuso**, nas quais o grupo de freqüências é único. Quando o padrão é repetido dentro do *cluster* as freqüências podem ser reutilizadas. Existem muitos tipos diferentes de *cluster*. A Figura 17.2 ilustra dois deles.

As células com mesmo número no *cluster* podem usar o mesmo conjunto de freqüências. Chamamos essas células de *co-células*. Como mostra a figura, no *cluster* com fator de reuso 4, uma única célula separa as células usando o mesmo grupo de freqüências. No *cluster* com fator de reuso 7, duas células separam as co-células.

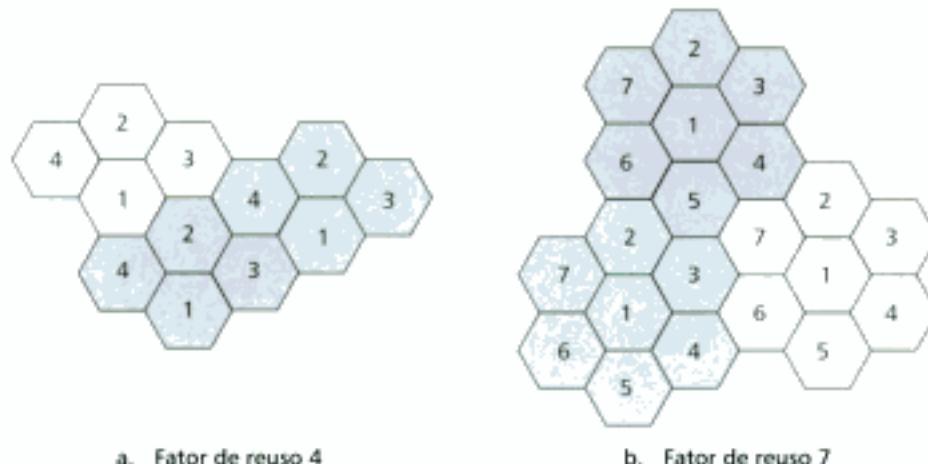


Figura 17.2 Padrões de reuso de freqüências.

Transmissão

Para realizar uma chamada, o usuário digita um código de 7 a 10 dígitos (o número de telefone) e pressiona a tecla *send*. A estação móvel (o celular) varre a banda procurando um canal com um nível de sinal forte e envia os dados (número de telefone) para a ERB mais próxima que estiver usando aquele canal. A ERB retransmite os dados à MSC. A MSC reenvia os dados à central telefônica responsável pelo controle e gerenciamento da ligação. Se a parte chamada estiver disponível, uma conexão é feita e o resultado é retransmitido de volta à MSC. Neste ponto, a MSC atribui um canal de voz disponível para a ligação e a conexão é estabelecida. As estações móveis ajustam-se automaticamente, buscando por novos canais noutras células para que a ligação continue.

Recepção

Quando uma chamada é dirigida a uma estação móvel, a central telefônica envia o número para a MSC. A MSC envia um sinal de consulta procurando a localização da estação móvel dentro de uma célula, processo esse denominado *paging*. Uma vez que a estação móvel é detectada, a MSC transmite um sinal de chamada e, quando a estação móvel responde, atribui um canal de voz dando início a comunicação de voz.

Handoff

Às vezes, acontece a migração da estação móvel de uma célula para outra. Quando isto ocorre, o sinal torna bastante enfraquecido. Para resolver este problema, a MSC monitora o nível de sinal em intervalos de poucos segundos. Se a intensidade do sinal decresce, a MSC procura uma nova célula que melhor mantenha a integridade da comunicação. Nesse caso, a MSC muda o canal que transporta a chamada (comutando o sinal do canal velho para o canal novo).

Hard Handoff Os sistemas mais antigos usavam o *hard handoff*. No *hard handoff*, uma estação móvel comunica-se somente com uma única estação base. Quando a estação móvel desloca-se de uma célula para a outra, a comunicação é quebrada primeiramente com a estação base antiga antes da comunicação ser estabelecida com a nova. Isto pode criar uma transição abrupta entre ERBs.

Soft Handoff Os novos sistemas usam o *soft handoff*. Neste caso, uma estação móvel pode se comunicar com duas estações base ao mesmo tempo. Isto significa que, durante o *handoff*, uma estação pode iniciar a transmissão para a nova ERB antes de romper com a velha.

Roaming

Roaming é uma característica importante da telefonia celular de prover à estação móvel comunicação fora de sua área nativa de serviço. Uma operadora geralmente possui cobertura limitada. As operadoras na vizinhança podem estender suas respectivas coberturas através de um contrato de *roaming*. A situação é similar ao serviço de correio entre dois países.

Primeira Geração

A telefonia celular está nesse momento na segunda geração com a terceira já despontando no horizonte. A primeira geração da telefonia celular foi projetada para comunicação de voz através de canais analógicos. Vamos analisar o sistema de comunicação móvel de primeira geração utilizado nos Estados Unidos e em boa parte dos países: o sistema AMPS*.

AMPS

A sigla **AMPS** significa **Advanced Mobile Phone System**. Ela foi a tecnologia de primeira geração dominante para serviços de telefonia celular analógica nos Estados Unidos. Esse sistema separava os canais no *link* através da multiplexação FDMA.

AMPS é um sistema de telefonia celular que utiliza a multiplexação FDMA.

Bandas O sistema AMPS opera numa faixa de freqüência ISM de 800 MHz. O sistema usa dois canais analógicos separados para comunicação de ERB para estação móvel e vice-versa. A banda entre 824 e 849 MHz é chamada banda A e a banda compreendida entre 869 e 894 MHz é a banda B (veja a Figura 17.3).

* N. de R. T.: O sistema AMPS foi adotado também no Brasil e ainda existem resquícios dele por aqui.

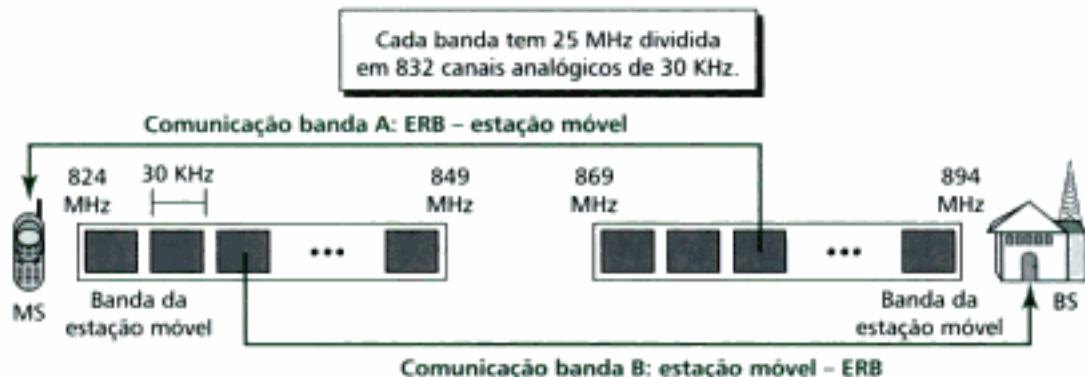


Figura 17.3 Bandas do padrão AMPS.

Cada banda foi dividida em 832 canais. Entretanto, uma região era compartilhada entre duas operadoras, o que representava 416 canais em cada célula por operadora. Desse total (416 canais), 21 eram utilizados para controle deixando 395 canais disponíveis para comunicação. O sistema AMPS tinha um fator de reuso de freqüência igual a 7. Isto quer dizer que apenas 1/7 destes 395 canais estavam realmente disponíveis numa célula.

Transmissão O sistema AMPS usa modulação FM e FSK. A Figura 17.4 mostra a transmissão na banda B. Os canais de voz são modulados usando FM e os canais de controle usam FSK para criar sinais analógicos de 30 KHz. A divisão dos 25 MHz da banda em canais de 30 KHz era feita através da multiplexação FDMA.

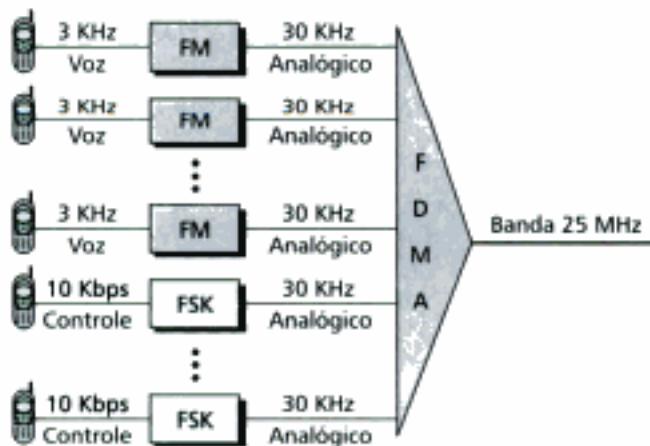


Figura 17.4 Comunicação AMPS banda B.

Segunda Geração

A segunda geração da telefonia celular foi desenvolvida para proporcionar comunicação de voz de alta qualidade (menos propenso aos ruídos). Enquanto a primeira geração tinha sido desenvolvida para suportar canais de voz analógicos, a segunda geração foi projetada principalmente para suportar canais de voz digitalizados. Os três grandes sistemas desenvolvidos para segunda geração são mostrados na Figura 17.5. Analisaremos cada um desses sistemas em separado.

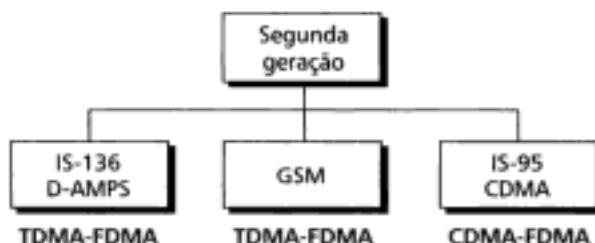


Figura 17.5 Sistemas de telefonia celular de segunda geração.

D-AMPS

Este padrão é a evolução natural do padrão AMPS analógico. O sistema **Digital AMPS (D-AMPS)** foi desenvolvido de forma a manter compatibilidade total com o sistema AMPS mais antigo. Queremos dizer que numa célula, um telefone pode usar tanto AMPS quanto D-AMPS. A especificação D-AMPS foi desenvolvida primeiramente pelo IS-54 (*Interim Standard 54*) e revisado mais tarde pelo IS-136.

Banda O sistema D-AMPS usa as mesmas bandas e canais que o sistema AMPS analógico.

Transmissão Cada canal de voz é digitalizado através de uma codificação PCM complexa, aliada a uma técnica de compressão de dados eficiente. Um canal de voz é digitalizado a 7,95 kbps. Em seguida, três canais de voz digitalizados de 7,95 kbps são combinados através da técnica TDMA. O resultado disso são os dados digitalizados a 48,6 kbps, sendo boa parte dessa taxa dedicada à sinalização (*overhead*).

A Figura 17.6 ilustra o sistema transmitindo 25 frames por segundo, com 1944 bits por frame. Cada frame dura 40 ms (1/25) e é dividido em seis slots compartilhados por três canais digitais. Um canal é partilhado por dois slots.

Além disso, cada slot suporta 324 bits. Entretanto, somente 159 bits vêm da voz digitalizada; 64 bits são dedicados ao controle e 101 bits são reservados à correção de erros. Noutras palavras, cada canal transporta 159 bits de dados em cada um dos dois canais atribuídos a ele. O sistema adiciona 64 bits de controle e 101 bits de correção de erros.

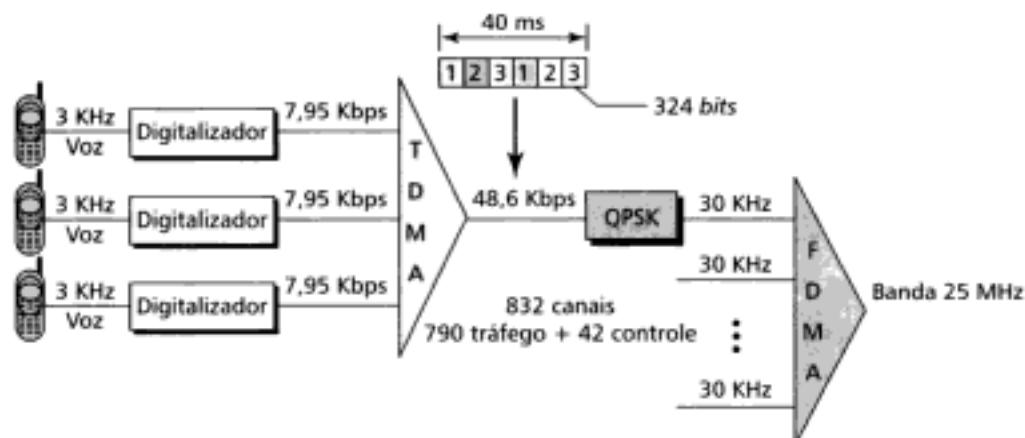


Figura 17.6 D-AMPS.

Os dados digitais a 48,6 kbps resultantes modulam uma portadora através da técnica QPSK. O resultado final é um canal analógico de 30 KHz. Finalmente, os sinais analógicos de 30 KHz são multiplexados em freqüência formando a banda de 25 MHz. O sistema D-AMPS tem fator de reuso de freqüência igual a 7.

D-AMPS ou IS-136 é um sistema de telefonia móvel celular usando TDMA e FDMA.

GSM

O sistema **Global System for Mobile Communication (GSM)** é um padrão europeu desenvolvido de modo a oferecer serviços de telefonia celular de segunda geração para toda a Europa. A idéia central por trás desse padrão era substituir uma certa quantidade de tecnologias incompatíveis de primeira geração.

Bandas O sistema GSM também utiliza duas bandas para comunicação *duplex*. Cada banda possui uma largura de faixa de 25 MHz, a menor centrada em aproximadamente 902 MHz e a maior centrada em 947 MHz, conforme ilustra a Figura 17.7. Cada banda é subdividida em 124 canais devidamente separados por faixas de segurança.

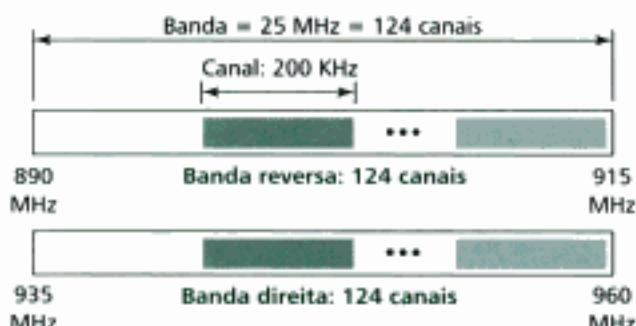


Figura 17.7 Bandas GSM.

Transmissão A Figura 17.8 mostra um sistema GSM típico. Os canais de voz são digitalizados e comprimidos para formar um sinal digital de 13 kbps. Em cada *slot-time* são transmitidos 156,25 bits. Oito slots são multiplexados de modo a formar um *frame* TDM. Em seguida, 26 frames são combinados para formar um *multiframe*. Podemos calcular a taxa de transmissão de cada canal da seguinte maneira:

$$\text{Taxa de transmissão do canal} = (1/120 \text{ ms}) \times 26 \times 8 \times 156,25 = 270,8 \text{ kbps}$$

Cada canal digital de 270,8 kbps é modulado numa portadora através de GMSK (uma variante da modulação FSK predominante em sistemas europeus). O resultado é um sinal analógico de 200 kHz. Finalmente, 124 analógicos de 200 kHz cada são multiplexados através da técnica FDMA. O resultado é uma banda de 25 MHz.

A Figura 17.9 ilustra os dados do usuário e a sinalização (*overhead*) dentro do *multiframe*.

O leitor deve ter percebido a enorme geração de *overhead* na etapa TDMA. Os dados dos usuários ocupam somente 65 bits dentro do slot. Num primeiro momento, o sistema adiciona bits extras de correção de erro e o tamanho do slot passa a 114 bits. Para tanto são adicionados bits de controle elevando a 156,25 bits por slot. Além disso, um frame encapsula 8 slots. Assim, o multiframe é formado por 24 frames de tráfego e dois frames adicionais de controle. Além disso, a arquitetura GSM também define os superframes e os hiperframes que não adicionam nenhum overhead. Não discutiremos tais frames neste livro.

Fator de Reuso O sistema GSM possui um fator de reuso 3 devido à complexidade do mecanismo de correção de erro.

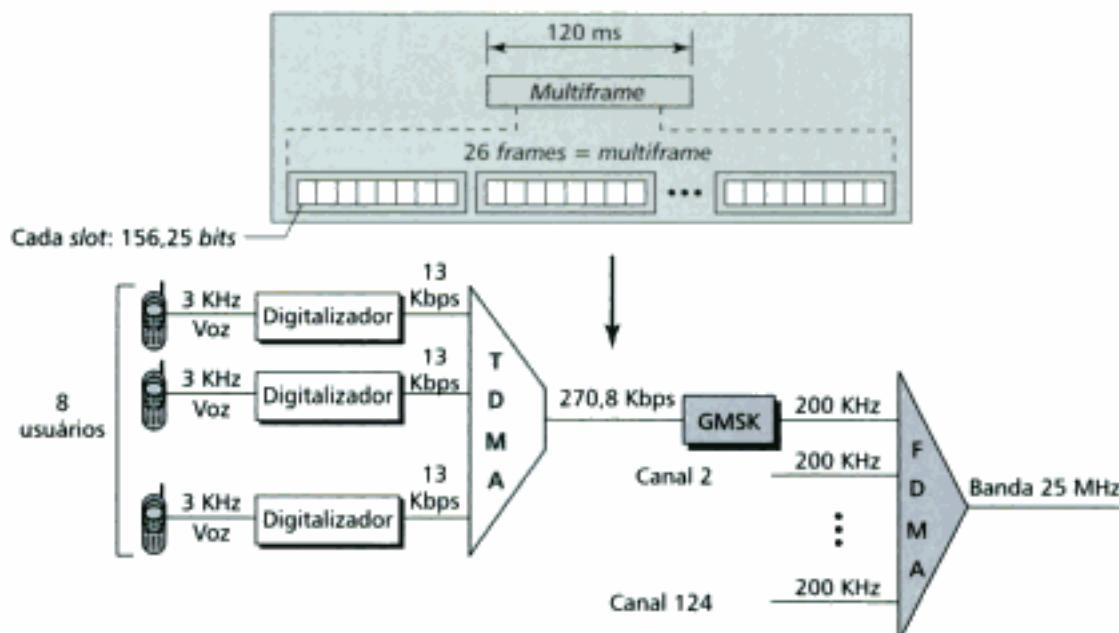


Figura 17.8 GSM.

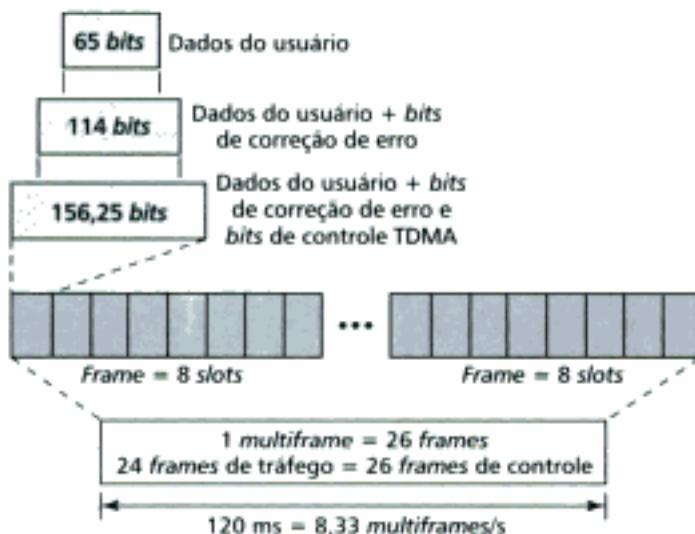


Figura 17.9 Componentes do multiframe.

GSM é um sistema de telefonia celular que utiliza TDMA e FDMA.

IS-95

Um dos padrões de segunda geração predominante nos Estados Unidos é o **Interim Standard 95 (IS-95)**. Esse padrão está baseado no CDMA e DSSS.

Bandas e Canais O IS-95 usa duas bandas para comunicação *duplex*. As bandas podem ser a faixa ISM tradicional de 800 MHz ou a faixa ISM de 1900 MHz. Cada banda é dividida em 20 canais de 1,228 MHz devidamente separados por bandas de segurança. Para cada operadora de tele-

fonia são alocados 10 canais. A IS-95 pode ser utilizada paralelamente ao sistema AMPS. Cada canal IS-95 equivale a 41 canais AMPS ($41 \times 30 \text{ kHz} = 1,23 \text{ MHz}$).

Sincronização Todos os canais base precisam estar sincronizados de modo a utilizar o CDMA. Para sincronização das ERBs é utilizado o serviço GPS (Global Positioning System), sistema de posicionamento via satélite que será discutido na próxima seção.

Transmissão direta O IS-95 possui duas técnicas de transmissão diferentes: uma para uso na comunicação direta (estação móvel–ERB) e outra para comunicação reversa (ERB–estação móvel). Nas comunicações entre uma ERB e todas as estações móveis sob domínio dela, há uma forte necessidade de sincronização. Isto porque as ERBs transmitem dados sincronizados para todas as estações móveis. A Figura 17.10 ilustra um diagrama simplificado para a comunicação direta.

Cada canal de voz é digitalizado produzindo dados a uma taxa básica de 9,6 kbps. Após a adição dos bits de correção de erro, repetição e de separação, o resultado é um sinal de 19,2 kbps (kilosinal por segundo). Esse sinal de saída é randomizado através de outro sinal 19,2 kbps (oriundo do bloco decimador). O sinal randomizado é produzido a partir de um gerador de código que utiliza um número eletrônico serial (Electronic Serial Number – ESN) da estação móvel e gera 2^{42} chips codes pseudo-aleatórios, cada qual com 42 bits de código. Perceba que esses chips codes são gerados pseudo-aleatoriamente, não aleatoriamente, visto que o padrão em si é repetitivo. A saída do gerador de código alimenta o bloco decimador, o qual seleciona 1 dentre 64 bits representando cada canal. Por último, a saída do decimador é utilizada como randomizadora do sinal de voz digitalizado. A randomização é utilizada para gerar privacidade e o número ESN é único para cada estação.

O resultado do randomizador alimenta o multiplexador CDMA. Para cada canal de tráfego é selecionada uma linha da matriz de Walsh de dimensão 64×64 . O resultado é um sinal de 1,288 Mcps (megachips por segundo).

$$19.2 \text{ kbps} \times 64 \text{ cps} = 1.288 \text{ Mcps}$$

O sinal multiplexado em CDMA alimenta o modulador QPSK de modo a produzir um sinal de 1.288 MHz. A largura de banda resultante (25 MHz) é gerada através da multiplexação FDMA.

Partindo de um canal analógico são criados 64 canais digitais, dos quais 55 são canais de tráfego (transportando voz digitalizada). Nove canais são utilizados para controle e sincronização:

- O canal 0 é o canal piloto. Nesse canal é enviado uma sequência contínua de 1s às estações móveis. Essa sequência proporciona sincronização bit a bit, serve como referência de

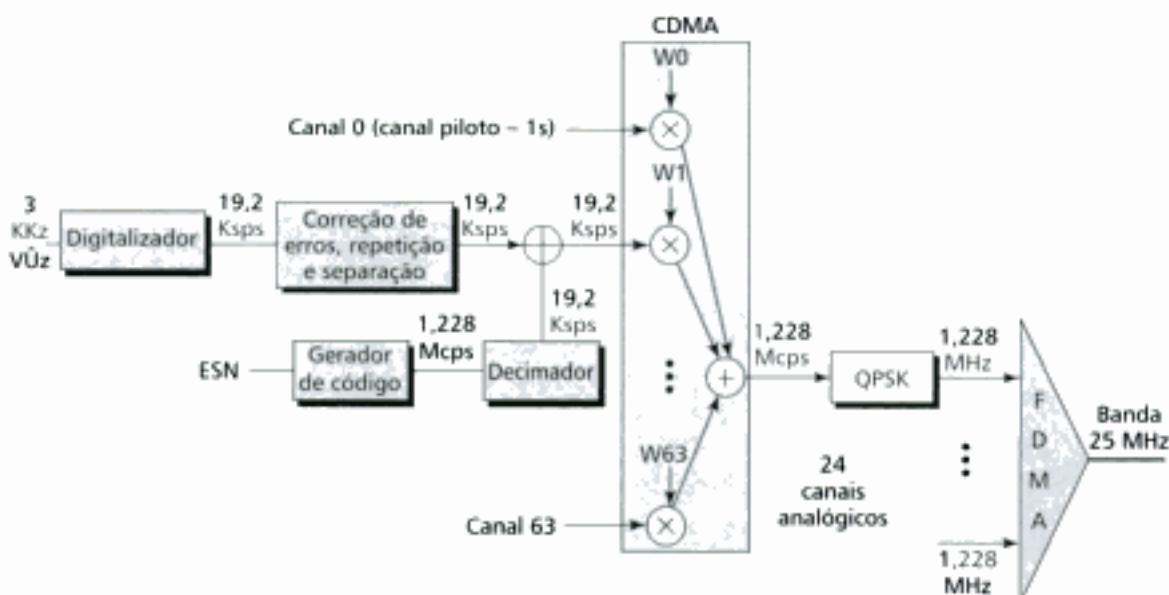


Figura 17.10 Transmissão direta no sistema IS-95.

fase para demodulação e permite à estação móvel comparar a intensidade do sinal numa ERB, relativamente às ERBs vizinhas, para tomada de decisão de *handoff*.

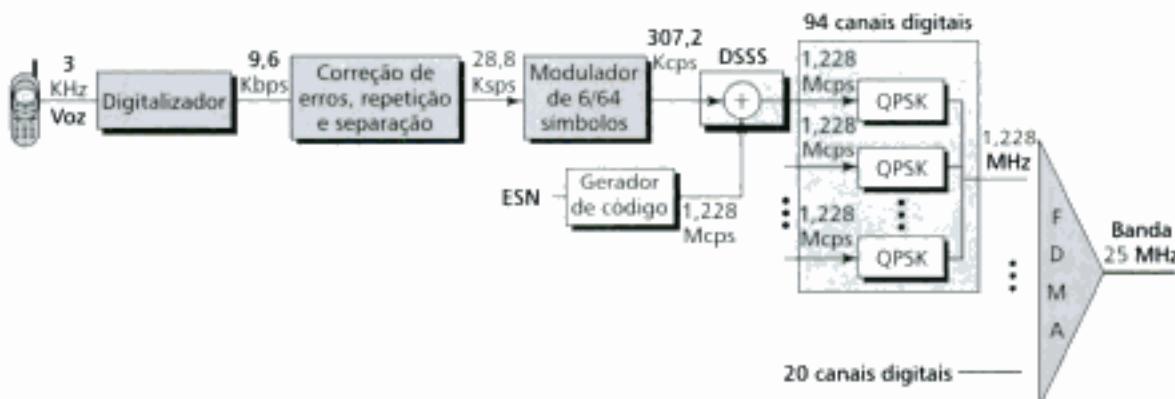
- O canal 32 transmite às estações móveis informação sobre o sistema.
- Os canais 1 a 7 são utilizados no *paging*, para transmitir mensagens de uma ou mais estações móveis.
- Os canais 8 a 31 e 33 a 63 são canais reservados ao tráfego para transporte de voz digitalizada entre a ERB e a estação móvel correspondente.

Transmissão Reversa A utilização do CDMA na transmissão direta é possível graças aos esforços do canal piloto que envia a seqüência contínua de 1s para sincronização das transmissões. Nenhuma sincronização é utilizada na transmissão reversa visto que necessitamos de uma entidade para fazer isso, o que não é plausível. Em vez de CDMA, os canais reversos usam DSSS (Direct Sequence Spread Spectrum), discutido no Capítulo 15. A Figura 17.11 mostra um diagrama simplificado para transmissão reversa.

Cada canal de voz é digitalizado produzindo dados numa taxa de 9,6 kbps. Entretanto, após a adição dos *bits* de correção de erro, repetição e de separação, o sinal resultante sai a 28,8 ksps. Em seguida, a saída passa através do modulador de 6/64 símbolos. Os símbolos são divididos em grupos (*chunks*) de seis símbolos e cada um deles é interpretado como um número binário (de 0 a 63). Os números binários são utilizados como índice da matriz de Walsh (dimensão 64×64) para seleção da linha do *chip code* apropriado. Perceba que este procedimento difere muito do CDMA pois cada *bit* não é multiplicado pelo *chip code* na linha. Assim, cada grupo de seis símbolos é substituído por um 64-*chip code*. Isto é necessário de modo a manter a ortogonalidade das seqüências e permitir ao sistema diferenciar os *chips* das diferentes estações móveis sob sua tutela. Este procedimento cria um sinal resultante a 307,2 kcps ou $(28,8/6) \times 64$.

A próxima etapa acontece no DSSS, onde cada *chip code* é espalhado (*spreading*) em 4. Novamente, o ESN da estação móvel cria um código de 42 *bits* a uma taxa de 1,228 Mcps, resultado do produto 4 vezes 307,2. Após o DSSS cada sinal é modulado através de QPSK, o que difere muito pouco daquele usado na transmissão direta e por isso não entraremos em detalhes sobre ele. Perceba que o diagrama não menciona o mecanismo de acesso múltiplo ao meio. Todos os canais reversos transmitem os respectivos sinais analógicos através do ar, mas a seqüência correta de *chips* será recebida pela ERB devido ao fenômeno de espalhamento (*spreading*).

Embora o gerador de código permita criar $2^{12} - 1$ canais digitais diferentes, apenas 94 canais são utilizados normalmente; 62 são canais de tráfego e 32 são canais utilizados pelas estações móveis para obtenção de acesso junto as ERBs.



O IS-95 é um sistema de telefonia digital que utiliza CDMA/DSSS e FDMA.

Dois Grupos de Taxa de Transmissão O IS-95 define dois grupos de taxas de transmissão, cada um deles contendo quatro diferentes taxas. O primeiro grupo define as taxas 9.600, 4.800, 2.400 e 1.200 bps. Se, por exemplo, a taxa seleciona for 1200 bps, cada *bit* é repetido 8 vezes de modo a proporcionar uma taxa de 9600 bps. No segundo grupo encontramos as taxas de 14.400, 7.200, 3.600 e 1800 bps. Isto possibilita reduzir a quantidade de *bits* utilizados na correção de erros. As taxas de transmissão estão relacionadas à atividade do canal. Se o canal estiver em "silêncio", somente 1.200 *bits* podem ser transferidos, o que melhora o espalhamento (*spreading*) pela repetição (8 vezes) de cada *bit*.

Fator de Reuso de Freqüência Normalmente, num sistema IS-95, o fator de reuso de freqüência é unitário visto que a interferência das células vizinhas não pode afetar as transmissões CDMA ou DSSS.

Soft Handoff Todas as ERB transmitem sinais de *broadcast* continuamente através do canal piloto. Isto significa que uma estação móvel pode detectar o sinal piloto de sua célula e das demais células vizinhas. Isto habilita a estação móvel a fazer *soft handoff* em vez de *hard handoff*.

PCS

Antes de finalizarmos a discussão sobre os sistemas de telefonia de segunda geração, vamos explicar um termo muito difundido associado à essa geração: PCS. O sistema **PCS (Personal Communications System)** não se refere a uma única tecnologia tal como GSM, IS-136 ou IS-95. Na verdade PCS é o nome genérico para um sistema comercial que oferece diversos tipos de serviços de comunicação. Podemos sintetizar as características em comum desses sistemas:

1. Podem utilizar qualquer tecnologia de segunda geração (GSM, IS-136 ou IS-95).
2. Utilizam a faixa de freqüências ISM de 1900 MHz. Significa que as estações móveis necessitam de maior potência de transmissão porque as freqüências mais altas têm um alcance mais curto, relativamente às freqüências menores. Entretanto, visto que a potência das estações móveis é limitada pelo FCC e demais órgãos competentes, há necessidade da estação base (ERB) estar mais próxima das estações móveis (isto é, as células são menores).
3. Oferecem serviços de comunicação tais como Short Message Service (SMS) e acesso limitado à Internet.

Terceira Geração

A terceira geração da telefonia celular refere-se a uma combinação de tecnologias que proporcionam uma variedade de serviços. Idealmente, quando estiver disponível, a terceira geração proporcionará tanto comunicação de dados como comunicação de voz digitalizada. Através de um pequeno dispositivo móvel, um usuário será capaz de falar em qualquer parte do mundo com uma qualidade de voz similar à existente na rede de telefonia fixa. O usuário também poderá fazer *downloads* e assistir filmes, ouvir músicas, navegar ou jogar na Internet, assistir a uma vídeo conferência e muito mais. Uma das características interessantes da terceira geração é que os dispositivos portáteis estarão sempre conectados, isto é, não será necessário discar um número para estar conectado à Internet.

O conceito da tecnologia de terceira geração surgiu em 1992, quando o ITU publicou uma proposta de projeto denominada **Internet Mobile Communication for year 2000 (IMT-2000)**. O documento define alguns critérios para a tecnologia 3G:

- Qualidade de voz comparável ao existente nos serviços de telefonia fixo.
- Taxas de transmissão de dados de 144 kbps para acesso a partir de veículo móvel (carros), 384 kbps para acesso partindo de usuário em movimento (pedestres) e 2 Mbps para usuários fixos (casa ou escritório).

- Suportar serviços de comutação de pacotes e comutação de circuitos.
- Banda total de 2 GHz.
- Largura de banda de 2 MHz.
- Interface com a Internet.

O objetivo principal da telefonia celular de terceira geração é oferecer serviços de comunicação universal, isto é, sem restrições às pessoas.

Interface de Rádio do Padrão IMT-2000

A Figura 17.12 mostra as interfaces de rádio (padrão *wireless*) adotadas pelo IMT-2000. Todas as cinco interfaces foram desenvolvidas a partir das tecnologias de segunda geração. As duas primeiras evoluíram da tecnologia CDMA. A terceira evoluiu simultaneamente da combinação CDMA e TDMA. A quarta evoluiu da TDMA e a última evoluiu tanto da FDMA quanto da TDMA.

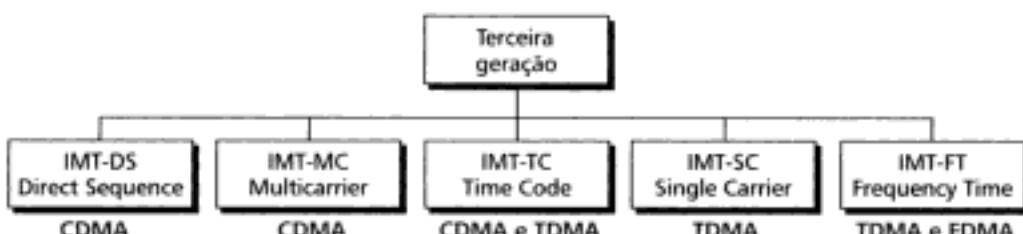


Figura 17.12 Interfaces de rádio do padrão IMT-2000.

IMT-DS Este padrão utiliza uma versão CDMA denominada CDMA banda larga ou W-CDMA. O W-CDMA utiliza uma largura de banda de 5 MHz. Ele foi desenvolvido na Europa e é compatível com o CDMA utilizado pelo IS-95.

IMT-MC Este padrão foi desenvolvido nos Estados Unidos e é conhecido pela sigla CDMA 2000. Ele é uma evolução da tecnologia CDMA utilizada nos canais IS-95. Além do mais, combina a banda larga *spread spectrum* em 15 MHz com a banda estreita de 1,25 MHz utilizada na tecnologia IS-95. Portanto, esse padrão é totalmente compatível com o IS-95. Além disso, permite comunicação em canais múltiplos de 1,25 MHz (1, 3, 6, 9, 12 vezes) até 15 MHz. A utilização de canais maiores permite atender a taxa de transmissão de 2 Mbps exigida pela terceira geração da telefonia celular.

IMT-TC Este padrão usa uma combinação W-CDMA e TDMA. O padrão se propõe a alcançar as metas do IMT-2000 adicionando multiplexação TDMA ao W-CDMA.

IMT-SC Este padrão usa somente TDMA.

IMT-FT Este padrão usa uma combinação FDMA e TDMA.

17.2 REDES DE SATÉLITE

Uma **rede de satélites** é uma combinação de nós organizados espacialmente de modo a prover comunicação de um ponto a outro sobre a superfície da Terra. Um nó numa rede pode ser um satélite artificial, uma estação fixa na Terra ou um usuário final de terminal ou telefone via satélite. Embora um satélite natural, tal como a Lua, possa ser utilizado como um nó na rede, o uso de satélites artificiais é preferido porque neles instalamos equipamentos eletrônicos para regenerar os sinais que invariavelmente perdem energia durante a viagem. Outra restrição quanto à utilização de

satélites naturais é que as distâncias desses corpos relativamente à Terra é muito grande e, por isso, provocam muitos atrasos (*delays*) na comunicação.

As redes de satélites funcionam de modo bastante semelhante às redes de telefonia móvel. Elas dividem o planeta em grandes células. Os satélites conseguem atingir quaisquer pontos sobre a Terra, não importando o quanto remoto estejam. Esta vantagem torna possível a comunicação com as partes mais longínquas sobre a Terra com relativamente pouco investimento em infra-estrutura baseada em solo.

Órbitas

Um satélite artificial deve ser colocado em **órbita** ao redor da Terra. A órbita de um satélite pode ser equatorial, inclinada ou polar, conforme ilustra a Figura 17.13.

O período de um satélite é o tempo necessário para que o satélite dê uma volta completa em torno da Terra. O período é determinado pela lei de Kepler, a qual define o período como uma função da distância do satélite ao centro da Terra.

$$\text{Período} = C \times \text{distância}^{1.5}$$

Onde C é uma constante aproximadamente igual a 1/100. Nessa fórmula, o período é dado em segundos enquanto a distância é dada em quilômetros.

Exemplo 1

De acordo com a lei de Kepler, qual é o período de revolução da lua?

Solução

A lua está localizada a aproximadamente 384.000 km da superfície da Terra. O raio da Terra vale 6378 km. Aplicando a fórmula de Kepler chegamos a:

$$\text{Período} = (1/100) (384.000 + 6378)^{1.5} \rightarrow 2.439.090\text{s} \rightarrow 1 \text{ mês}$$

Exemplo 2

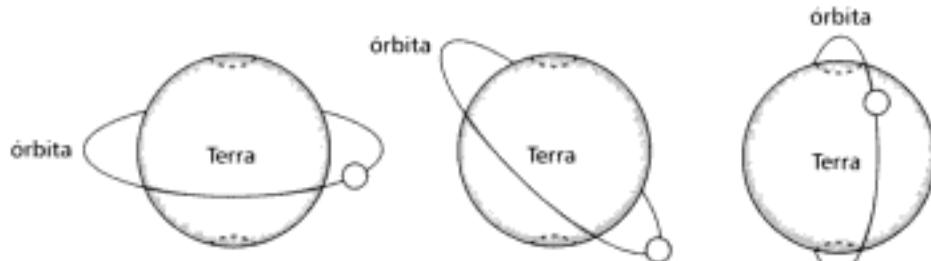
De acordo com a lei de Kepler, qual o período de um satélite localizado numa órbita de aproximadamente 35.786 km acima da Terra?

Solução

Aplicando a fórmula, temos:

$$\text{Período} = (1/100) (35.786 + 6378)^{1.5} \rightarrow 86.579\text{s} \rightarrow 24 \text{ h}$$

Isto significa que um satélite artificial localizado a 35.786 km tem um período de 24 h, o que corresponde ao período de rotação da Terra. Um satélite como este é denominado *estacionário* relativamente à Terra. A órbita, como veremos, é denominada geoestacionária.



a. Satélite em órbita equatorial b. Satélite em órbita inclinada c. Satélite em órbita polar

Figura 17.13 Órbitas de satélites.

Footprint

Os satélites realizam transmissões em microondas através de antenas bidirecionais. Assim, o sinal oriundo de um satélite cobre uma área cônica específica sobre a Terra denominada *footprint*. A potência do sinal no centro do *footprint* é máxima. A potência decresce à medida que nos movemos do centro em direção à borda do cone. Na borda do *footprint* estão localizados os pontos onde a potência atinge um limiar predeterminado.

Três Categorias de Satélites

Baseado na localização da órbita, os satélites podem ser divididos em três categorias: GEO, LEO e MEO. A Figura 17.14 mostra a taxonomia.

A Figura 17.15 mostra as altitudes dos satélites com respeito à superfície da Terra. Os satélites GEO só podem estar localizados a uma altitude de 35.786 km da Terra. Os satélites MEO estão localizados em altitudes entre 5.000 e 15.000 km. Os satélites LEO estão normalmente abaixo dos 2000 km.

Uma razão para as diferentes órbitas é devido à existência de dois cinturões de Van Allen. Um cinturão de Van Allen é uma camada que contém partículas carregadas. Um satélite orbitando em um desses cinturões poderia ser completamente destruído por partículas carregadas de alta energia. As órbitas MEO estão localizadas entre os dois cinturões.

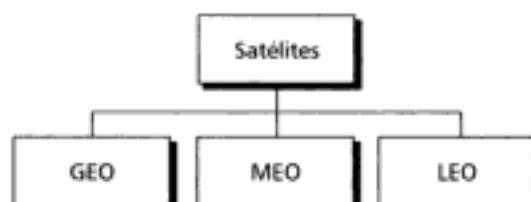


Figura 17.14 Categorias de satélites.



Figura 17.15 Altitudes das órbitas de satélites.

Bandas de Freqüência para Comunicação Via Satélite

As freqüências reservadas à comunicação via satélite estão na faixa de microondas, ou seja, ocupam alguns gigahertz (GHz) na faixa espectral. Cada satélite transmite e recebe em duas bandas diferentes. A transmissão da Terra ao satélite é denominada *uplink* (rota de subida). A transmissão do satélite à Terra é denominada *downlink* (rota de descida). A Tabela 17.1 mostra os nomes das bandas de freqüência e as respectivas freqüências de *uplink* e *downlink*.

TABELA 17.1 Bandas de freqüência de satélite

Banda	Downlink, GHz	Uplink, GHz	Largura de banda (MHz)
L	1,5	1,6	15
S	1,9	2,2	70
C	4	6	500
Ku	11	14	500
Ka	20	30	3500

Satélites GEO

A propagação direcionada requer que as antenas transmissoras e receptoras estejam focadas durante todo o tempo (uma antena deve sempre ter a outra em seu “campo de visão”). Por esta razão, os satélites que se movem mais rápidos ou mais lentos que a rotação da Terra são úteis apenas durante um curto período de tempo. Para assegurar comunicação constante, o satélite deve mover-se na mesma velocidade de rotação da Terra, tal que a posição do satélite relativamente aos pontos de cobertura sobre a superfície da Terra seja mantida fixa. Um satélite com essas características é denominado **geoestacionário**.

Visto que a velocidade orbital está baseada na distância ao planeta, existe apenas uma órbita geoestacionária. Esta órbita fica localizada no plano equatorial e está a aproximadamente 36.000 km acima da superfície da Terra.

Definitivamente, um único satélite geoestacionário não pode cobrir toda a superfície da Terra. Um satélite nessa órbita tem contato na linha de visão com muitas estações baseadas em solo, mas a curvatura do planeta cega a visão do satélite para muitos outros pontos em lados opostos da superfície. São necessários no mínimo três satélites equidistantes uns dos outros, em **órbitas geoestacionárias** ou **Geosynchronous Earth Orbit (GEO)**, para prover comunicação global à Terra. A Figura 17.16 mostra três satélites separados de 120° , em órbita geoestacionária, dispostos sobre a linha equador. A vista mostra o Pólo Norte.

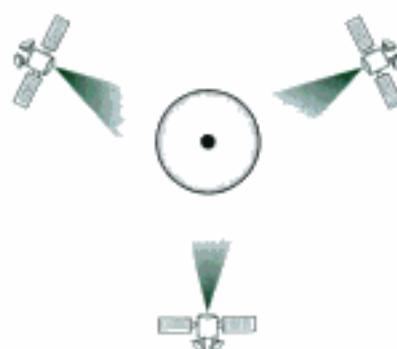


Figura 17.16 Satélites na órbita geoestacionária.

Satélites MEO

Os **satélites com órbitas de médias altitudes** ou **Medium-Earth Orbit (MEO)** estão posicionados entre dois cinturões de Van Allen. Um satélite nesta órbita leva aproximadamente 6 horas para circular a Terra.

GPS

Um exemplo importante de sistema de satélites MEO é o **GPS (Global Positioning System)** orbitando a uma altitude de 18.000 km acima da Terra. Embora o sistema GPS tenha sido colocado em órbita pelo Departamento de Defesa dos Estados Unidos, hoje ele é um sistema público. Este sistema é formado por 24 satélites e é utilizado para orientar a navegação terrestre e marítima, fornecendo posição espacial e temporal para veículos e navios. O GPS não é utilizado para comunicações.

A idéia que permeia o sistema GPS é a **triangulação**. Num plano, se soubermos nossa distância relativa a três pontos fixos saberemos exatamente aonde estamos. Por exemplo, vamos dizer que estamos a 10 km de um ponto A qualquer, 12 km do ponto B e 15 km do ponto C. Se desenharmos três círculos centrados em A, B e C deveremos estar localizados em algum lugar sobre o círculo A, B e C. Estes três círculos interceptam-se num único ponto: nossa localização (se as distâncias medidas dos pontos estiverem corretas). A Figura 17.17 ilustra o conceito. Entretanto, a localização sobre a superfície da Terra é espacial e a situação é um pouco diferente. Três esferas tocam-se em dois pontos. Por isso necessitamos de quatro esferas. Se soubermos nossa distância a partir de quatro pontos poderemos determinar exatamente onde estamos.

O sistema GPS usa 24 satélites em seis órbitas diferentes, como mostra a Figura 17.18. As órbitas e as localizações dos satélites em cada órbita são projetadas de tal maneira que, em qualquer tempo, quatro satélites estão visíveis de qualquer ponto da Terra. Um receptor GPS possui um almanaque que informa a posição atual do satélite. Assim, o aparelho envia um sinal aos quatro satélites e mede quanto tempo o sinal leva para retornar. Desse modo, o receptor determina sua posição sobre a Terra. Um receptor GPS também pode mostrar sua localização espacial sobre um mapa.

O GPS é intensivamente utilizado por forças militares. Na Guerra do Golfo, foram utilizados milhares de receptores GPS portáteis pelos soldados a pé, nos veículos terrestres e em helicópteros. Um uso mais inteligente do sistema acontece na navegação. Os proprietários de carros podem determinar a localização do carro num grande estacionamento por exemplo. Quando chegar ao veículo, o proprietário consulta um banco de dados na memória do automóvel procurando pelo caminho de retorno. Noutras palavras, o GPS dá a localização do automóvel e o banco de dados determina o caminho de retorno. Mencionamos na seção anterior que o sistema de telefonia celular IS-95 usa o sistema GPS para sincronizar as ERBs.

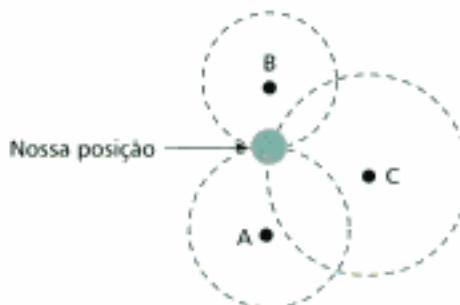


Figura 17.17 Triangulação.

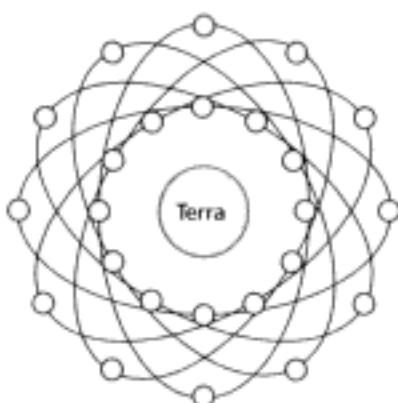


Figura 17.18 GPS.

Satélites LEO

Os **satélites de baixa altitude** ou **Low-Earth Orbit (LEO)** possuem órbitas polares. A altitude desses satélites gira entre 500 e 2000 km, com um período de rotação variando entre 90 e 120 minutos. Estes satélites orbitam em velocidades fantásticas, algo entre 20.000 a 25.000 km/h. Um sistema LEO possui um tipo de acesso celular similar ao sistema de telefonia móvel. O *footprint* desses satélites tem um diâmetro de 8000 km. Como os satélites LEO estão muito próximos da superfície terrestre, o atraso de propagação é normalmente menor que 20 ms, o que é aceitável nas comunicações de áudio.

Um sistema LEO é feito a partir de uma constelação de satélites trabalhando juntos numa rede; cada satélite age como um *switch*. Satélites muito próximos entre si estão conectados através de *links* inter-satélites (ISLs). Um sistema móvel comunica-se com satélite através de *link* de usuá-
rio ou User Mobile Link (UML). Um satélite pode se comunicar com uma estação terrestre fixa, denominada *gateway*, através de Gateway Link (GWL). A Figura 17.19 ilustra uma rede de satélites LEO típica.

Os satélites LEO podem ser divididos em três tipos: pequeno LEO, grande LEO e LEO banda larga. Os pequenos LEOs operam na faixa de 1 GHz. Eles são bastante utilizados para comunicação de mensagens em baixas velocidades. Os grandes LEOs operam na faixa situada entre 1 e 3 GHz. Os sistemas Globalstar e Iridium são exemplos de grandes LEOs. Os LEOs banda larga proporcionam comunicação similares às redes de fibra óptica. O primeiro sistema LEO banda larga foi o Teledesic.

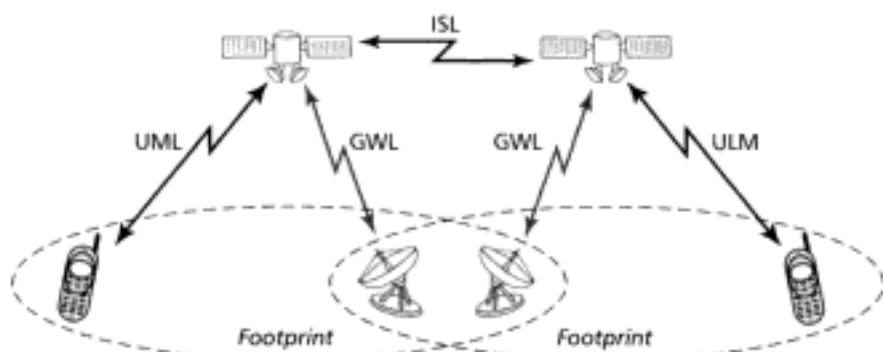


Figura 17.19 Sistema de comunicação dos satélites LEO.

Sistema Iridium

O conceito do sistema **Iridium**, uma rede de 77 satélites, foi proposto pela Motorola em 1990. O projeto levou 8 anos para ser materializado. Durante este período, a quantidade de satélites foi reduzida. Finalmente, em 1998 o serviço entrou em funcionamento com 66 satélites. O nome origi-

nal do sistema (Iridium) foi copiado do nome do 77º elemento químico. Um nome mais apropriado seria Disprósio (o nome do elemento número 66).

O Iridium passou por maus momentos. O sistema foi paralisado em 1999 devido a problemas financeiros, sendo vendido e reativado no ano de 2001 por um novo proprietário.

O sistema possui 66 satélites divididos em seis órbitas, isto é, com 11 satélites em cada órbita. Essas órbitas estão a uma altitude de 750 km. Os satélites em cada órbita estão separados de 32º de latitude. A Figura 17.20 mostra um diagrama esquemático da constelação Iridium.

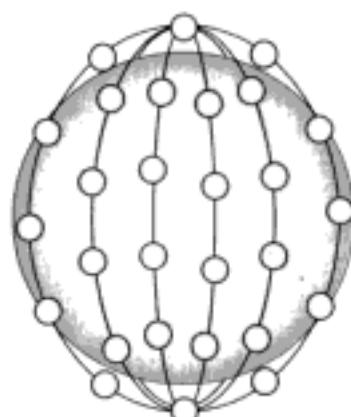


Figura 17.20 Constelação Iridium.

O sistema Iridium possui 66 satélites em 6 órbitas LEO de 750 km de altitude.

Como cada satélite possui 48 feixes de cobertura, todo o sistema tem cerca de 3168 feixes. Entretanto, alguns desses feixes são desligados quando o satélite aproxima-se do pólo. O número de feixes ativos no momento é aproximadamente 2000. Cada feixe cobre uma célula sobre a Terra, o que significa que a Terra foi dividida em aproximadamente 2000 células com sobreposição (*overlapping*).

No sistema Iridium, a comunicação entre dois usuários finais toma lugar através de *links* via satélite. Quando um usuário faz uma chamada, ela segue possivelmente através de muitos satélites antes de alcançar o destino. Isto significa que a comutação é feita no espaço e cada satélite necessita de um sistema de comutação bastante sofisticado. Esta estratégia elimina a necessidade de muitas estações terrestres.

O propósito do sistema Iridium é prover comunicação global direta através de terminais portáteis (o mesmo conceito da telefonia celular). O sistema pode ser utilizado para transmissão de voz, dados, *paging*, fax e até mesmo navegação. Este sistema consegue prover conectividade entre usuários em localizações onde outros tipos de comunicação não são possíveis. O sistema fornece transmissão de voz e dados entre telefones portáteis em taxas variando de 2,4 a 4,8 kbps. A transmissão acontece numa faixa de freqüência entre 1,616 e 1,626 GHz. Os *links* inter-satélites operam numa faixa de freqüências entre 23,18 e 23,38 GHz.

O sistema Iridium foi desenvolvido de modo a proporcionar comunicação global direta de voz e dados através de terminais portáteis. Um serviço similar à telefonia celular, mas em escala global.

Globalstar

O sistema **Globalstar** é outro exemplo de sistema LEO. Esse sistema usa 48 satélites em seis órbitas polares, onde cada órbita hospeda 8 satélites. As órbitas estão localizadas numa altitude de 1400 km.

O sistema Globalstar é semelhante ao sistema Iridium. A diferença principal é o mecanismo de comutação. No sistema Iridium, a comunicação entre dois usuários finais em pontos distantes sobre o globo requer comutação espacial entre os diversos satélites por onde ela passar. O sistema

Globalstar requer tanto comutação espacial quanto comutação em estações terrestres. Isto significa que as estações terrestres podem criar sinais mais potentes.

Teledesic

O sistema **Teledesic** é uma rede de satélites para prover comunicação semelhante às fibras ópticas (canais banda larga, com baixas taxas de erros e pequenos atrasos). O propósito principal é prover acesso à Internet em banda larga a usuários em todo o mundo. Ele é, às vezes, denominado "Internet no céu".

Este projeto iniciou em 1990 por Craig McCaw e Bill Gates. Posteriormente, outros investidores entraram no consórcio. O projeto está agendado para entrar em operação funcional definitiva em 2005.

Constelação O Teledesic é formado por 288 satélites em 12 órbitas polares, onde cada órbita hospeda 24 satélites. As órbitas estão a uma altitude de 1350 km, conforme ilustra a Figura 17.21.

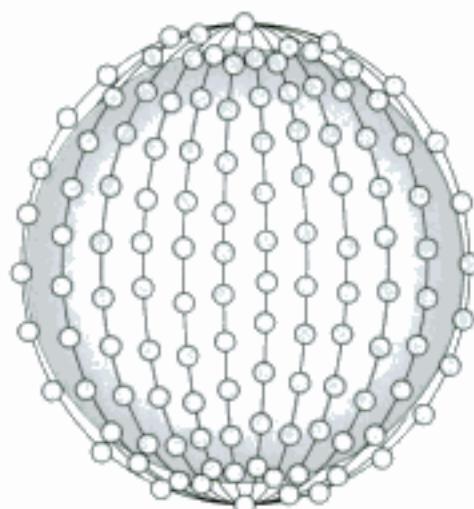


Figura 17.21 Teledesic.

O sistema Teledesic possui 288 satélites em 12 órbitas LEO de 1350 km de altitude.

Comunicação O sistema fornece três tipos de comunicação. A comunicação inter-satélites permite que oito satélites na vizinhança se comuniquem. Outra forma de comunicação acontece entre um satélite e uma estação terrestre (*gateway*). Os usuários finais podem se comunicar diretamente com a rede através de terminais. Além disso, a superfície da Terra foi dividida em cerca de 10.000 células. A cada célula é atribuído um *time-slot* e um satélite foca o seu feixe na célula correspondente ao *time-slot*. O terminal pode transmitir dados durante o respectivo *time-slot*. Um terminal recebe todos os pacotes destinados à célula onde ele está, mas seleciona somente aqueles destinados ao endereço dele.

Bandas As transmissões acontecem nas bandas Ka.

Taxas de Transmissão As taxas de transmissão previstas no projeto é 155 Mbps para o *uplink* e 1.2 Gbps para o *downlink*.

17.3 TERMOS-CHAVE

Advanced Mobile Phone System (AMPS)	Internet Mobile Communication for year 2000 (IMT-2000)
Central de Comutação e Controle	Iridium
Digital AMPS (D-AMPS)	Low-Earth Orbit (LEO)
<i>Downlink</i> (rota de descida)	Medium-Earth Orbit (MEO)
Fator de reuso	Órbita
<i>Footprint</i>	Padrão Interim 95 (IS-95)
Geosynchronous Earth Orbit (GEO)	Personal Communication System (PCS)
Global Positioning System (GPS)	Rede de satélites
Global System for Mobile Communication (GSM)	<i>Roaming</i>
<i>Globalstar</i>	Teledesic
<i>Handoff</i>	Telefonia celular
	Triangulação
	<i>Uplink</i> (rota de subida)

17.4 RESUMO

- A telefonia celular fornece serviços de comunicação entre dois dispositivos. Um dispositivo ou ambos pode ser móvel.
- Uma área de cobertura celular é dividida em células.
- O sistema Advanced Mobile Phone System (AMPS) fez parte da primeira geração de sistemas de telefonia celular.
- O sistema Digital AMPS (D-AMPS) é um sistema de telefonia celular de segunda geração que é a versão do padrão AMPS original.
- O sistema Global System for Mobile Communication (GSM) é um sistema de telefonia celular de segunda geração desenvolvido e utilizado na Europa.
- O padrão Interim 95 (IS-95) é um sistema de telefonia celular de segunda geração baseado em CDMA e DSSS.
- A terceira geração de sistemas de telefonia celular visam a comunicação pessoal universal.
- Uma rede de satélites utiliza satélites para fornecer serviços de comunicação entre quaisquer pontos sobre a superfície da Terra.
- O sistema de satélites Geosynchronous Earth Orbit (GEO) está localizado no plano equatorial e revoluciona em fase com a Terra.
- Global Positioning System (GPS) é um sistema de satélites de órbitas médias (MEO) que fornecem informação sobre a localização e o tempo para veículos e navios.
- Os satélites da constelação Iridium são do tipo Low-Earth Orbit (LEO) que proporcionam comunicação direta global de voz e dados para terminais portáteis.
- Os satélites da rede Teledesic são satélites LEO organizados de maneira a oferecer serviços de acesso à Internet banda larga.

17.5 PRATIQUE OS CONHECIMENTOS ADQUIRIDOS

Questões de Revisão

1. Qual é a relação entre uma estação rádio base (ERB) e uma central de comutação e controle (MSC)?
2. Quais são as funções de um MSC?
3. O que é melhor, um fator de reuso de frequência baixo ou alto? Explique sua resposta.
4. Qual é a diferença entre o *hard handoff* e o *soft handoff*?
5. O que significa AMPS?
6. Qual é a relação entre D-AMPS e AMPS?
7. O que é GSM?
8. Qual é a função do multiplexador CDMA no sistema IS-95?
9. Qual são os três tipos de órbitas?
10. Que tipo de órbita tem um satélite GEO? Explique sua resposta.
11. O que é *footprint*?
12. Qual é a relação entre os cinturões de Van Allen e os satélites?
13. Compare *uplink* e *downlink*.
14. Qual é o propósito do GPS?
15. Qual é a diferença principal entre os sistemas Iridium e Globalstar?

Questões de Múltipla Escolha

16. Uma _____ é um centro computadorizado responsável pela conexão das chamadas, registros das informações e tarifação.
- Estação base (ERB)
 - Central de comutação e controle
 - Célula
 - Estação móvel
17. Num(a) _____, uma estação móvel sempre se comunica com apenas uma ERB.
- Roaming*
 - Hard handoff*
 - Soft handoff*
 - Roaming handoff*
18. _____ é o nome da primeira geração de sistemas de telefonia celular.
- AMPS
 - D-AMPS
 - GSM
 - IS-95
19. _____ é uma segunda geração de sistemas de telefonia celular.
- D-AMPS
 - GSM
 - IS-95
 - Todas as opções acima
20. O sistema AMPS faz modulação _____.
- FM
 - FSK
 - PM
 - (a) e (b)
21. _____ separa os canais de voz AMPS.
- CDMA
 - TDMA
 - FDMA
 - (b) e (c)
22. _____ é um sistema de telefonia celular popular na Europa.
- AMPS
 - D-AMPS
 - GSM
 - IS-95
23. D-AMPS usa _____ para multiplexação.
- CDMA
 - TDMA
 - FDMA
 - (b) e (c)
24. GSM usa _____ para multiplexação.
- CDMA
 - TDMA
 - FDMA
 - (b) e (c)
25. DSSS é utilizado pelo sistema de telefonia celular _____.
- AMPS
 - D-AMPS
 - GSM
 - IS-95
26. As estações base no sistema _____ usam GPS para sincronização.
- AMPS
 - D-AMPS
 - GSM
 - IS-95
27. O IS-95 possui um fator de reuso de freqüência de _____.
- 1
 - 5
 - 7
 - 95
28. O percurso que um satélite desenvolve ao redor da Terra é denominado _____.
- Período
 - Footprint*
 - Órbita
 - Uplink*
29. Um satélite GEO possui uma órbita _____.
- Equatorial
 - Polar
 - Inclinada
 - Eqüilateral
30. O sinal oriundo de um satélite cobre uma área cônica específica sobre a Terra denominada _____.
- Período
 - Footprint*
 - Órbita
 - Uplink*
31. Que órbita possui a maior altitude?
- GEO
 - MEO
 - LEO
 - HEO
32. Os satélites MEO situam-se _____ cinturões de Van Allen.
- Nos
 - Entre os
 - Acima dos
 - Abaixo dos

33. A transmissão da Terra ao satélite é denominada _____.
a. *Footprint*
b. *Up link*
c. *Downlink*
d. *Uplink*
34. O sistema _____ não é utilizado para comunicação de voz.
a. IS-95
b. Globalstar
c. GPS
d. Iridium
35. _____ é freqüentemente utilizado para propósitos de navegação.
a. AMPS
b. IS-95
c. Iridium
d. GPS
36. Um satélite LEO possui uma órbita _____.
a. Equatorial
b. Inclinada
c. Polar
d. Todas as opções acima
37. Teledesic é um sistema de satélites de LEOs.
a. Pequenos
b. Grandes
c. Passa-banda
d. Banda larga
38. _____ possui 66 satélites em seis LEOs.
a. Globalstar
b. Iridium
c. Teledesic
d. GPS
39. _____ possui 48 satélites em seis órbitas polares.
a. Globalstar
b. Iridium
c. Teledesic
d. GPS
40. _____ possui 288 satélites em 12 órbitas polares.
a. Globalstar
b. Iridium
c. Teledesic
d. GPS

Exercícios

41. Desenhe uma célula padrão com fator de reuso de freqüência 3.
42. Qual é a quantidade máxima de chamadas simultâneas em cada célula AMPS?
43. Qual é a quantidade máxima de chamadas simultâneas em cada célula no sistema IS-136 (D-AMPS)?
44. Qual é a quantidade máxima de chamadas simultâneas em cada célula GSM?
45. Qual é a quantidade máxima de chamadas simultâneas em cada célula do sistema IS-95?
46. Qual é a eficiência do sistema AMPS em termos das chamadas por megahertz de largura de banda?
47. Qual é a eficiência do sistema D-AMPS em termos das chamadas por megahertz de largura de banda?
48. Qual é a eficiência do sistema GSM em termos das chamadas por megahertz de largura de banda?
49. Qual é a eficiência do sistema IS-95 em termos das chamadas por megahertz de largura de banda?
50. Deduza a relação entre um canal de voz de 3 kHz e um canal modulado a 30 KHz num tronco usando AMPS.
51. Quantos *slots* são enviados a cada segundo num sistema D-AMPS? Quantos *slots* são enviados por cada usuário nos 1s?
52. Você pode explicar por que a taxa de transmissão básica do sistema GSM é somente 13 kbps?
53. No sistema IS-95, quantos canais digitais estão disponíveis em cada célula?
54. O que acontece se um satélite for colocado acima da órbita GEO?
55. Use a lei de Kepler para verificar a precisão do período e da altitude para um satélite GPS.
56. Use a lei de Kepler para verificar a precisão do período e da altitude para um satélite Iridium.
57. Use a lei de Kepler para verificar a precisão do período e da altitude para um satélite Globalstar.

Comutação de Circuitos Virtuais: Frame Relay e ATM

Este é nosso último capítulo sobre a camada de enlace. Nos primeiros capítulos da parte III introduzimos os mecanismos de controle de fluxo e de erros e alguns protocolos da camada de enlace, tal como o HDLC. Explicamos os mecanismos de acesso múltiplos em redes LAN cabeadas e sem fios. Nesse ponto precisamos discutir uma última questão: comutação em WANs.

Introduziremos o conceito de **comutação de circuitos virtuais** como uma técnica de comutação utilizada nas redes WANs. Mostraremos como este tipo de comutação difere da comutação na camada física.

Atualmente temos duas tecnologias WAN bastante comuns utilizadas na comutação de circuitos virtuais. A tecnologia Frame Relay é um protocolo relativamente veloz que oferece serviços não disponíveis em outras tecnologias WAN tais como a DSL, TV a cabo e/ou tributários T ou E.

O protocolo ATM pode controlar uma *superhighway* de comunicação de dados quando o nível físico utilizado for especialmente rápido, como uma SONET por exemplo.

18.1 COMUTAÇÃO DE CIRCUITOS VIRTUAIS

No Capítulo 8 discutimos a comutação de circuitos. A comutação de circuitos é muito utilizada na camada física para o estabelecimento de circuitos *reais*, linhas dedicadas entre a origem de dados e o destino. Os circuitos reais foram especialmente desenvolvidos para transmitir sinais de áudio em tempo real (telefonia). Para a comunicação de dados, foram desenvolvidas as redes de comutação de pacotes cujo objetivo é encapsular dados e transmiti-los pacote por pacote. A principal diferença entre as redes de comutação de circuitos e comutação de pacotes é que na última os *links* são compartilhados, isto é, são canalizados entre os diferentes caminhos de comunicação. Um *link* entre os nós de comutação 1 e 2 pode transportar diversos tipos de pacotes ao mesmo tempo, cada qual enviado por fontes diferentes e direcionados para destinos diferentes.

A comutação de pacotes utiliza duas técnicas diferentes: os circuitos virtuais e os datagramas. A técnica de datagramas é desempenhada freqüentemente na camada de rede. Assim, postergaremos esta discussão até os últimos capítulos da Parte 4 deste livro. Os circuitos virtuais são uma tecnologia do nível de enlace e, por isso, serão tratados neste capítulo.

A Figura 18.1 mostra um exemplo de circuito virtual através de uma rede WAN. A rede é composta de nós de comutação que permitem o tráfego entre as origens de dados e os respectivos destinos. Uma origem/destino de dados pode ser um computador, um roteador, uma bridge ou qualquer dispositivo que conecte outras redes à rede WAN (LANs, por exemplo).



Figura 18.1 Circuito virtual WAN.

Endereçamento Global

Tanto a origem como o destino dos dados devem ser identificados através de um endereço global, ou seja, um endereço único no escopo da WAN particular, ou então, no nível internacional, caso a WAN seja utilizada como parte de uma rede mundial. Entretanto, veremos que o esquema de endereçamento global nas redes de circuitos virtuais é utilizado somente para criar um identificador de circuito virtual, como será visto na próxima seção.

Identificador de Circuito Virtual

O identificador utilizado de fato nas transferências de dados é denominado **identificador de circuitos virtuais** ou **Virtual Circuit Identifier (VCI)**. Diferentemente do endereço global, o VCI é um número utilizado somente na esfera da comutação. Ele é utilizado pelo *frame* enviado entre dois nós de comutação. Quando um *frame* chega ao nó de comutação, ele já recebeu a identificação VCI numa etapa anterior, antes de chegar ao nó. Quando o *frame* deixa o nó de comutação o número é outro. A Figura 18.2 ilustra como o identificador VCI é modificado num *frame* de dados ao passar pelo nó de comutação.

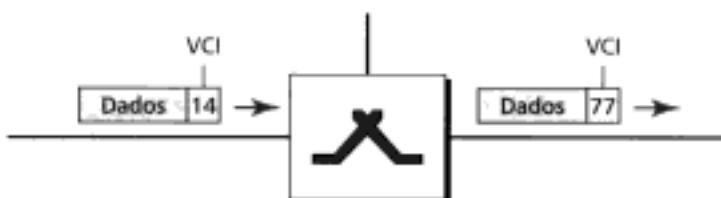


Figura 18.2 VCI.

Perceba que o identificador VCI não precisa ser um número muito grande já que cada nó de comutação pode utilizar seu próprio conjunto de identificadores.

Comunicação em Três Fases

A comunicação via comutação de circuitos virtuais entre a origem de dados e o destino deve acontecer em três fases: **estabelecimento do circuito, transferência da informação e desconexão do circuito** (veja a Figura 18.3).

Na fase de estabelecimento do circuito, a origem e o destino utilizam os respectivos endereços globais para auxiliar os nós de comutação na construção de entradas nas tabelas de comutação. Na fase de desconexão do circuito, a origem e o destino solicitam a limpeza das correspondentes entradas nas tabelas de comutação. A fase de transferência da informação (dados) acontece entre essas duas fases. Examinaremos primeiramente a fase de transferência de dados porque ela é relativamente simples e direta. As fases de estabelecimento e desconexão dos circuitos são explicadas logo em seguida.

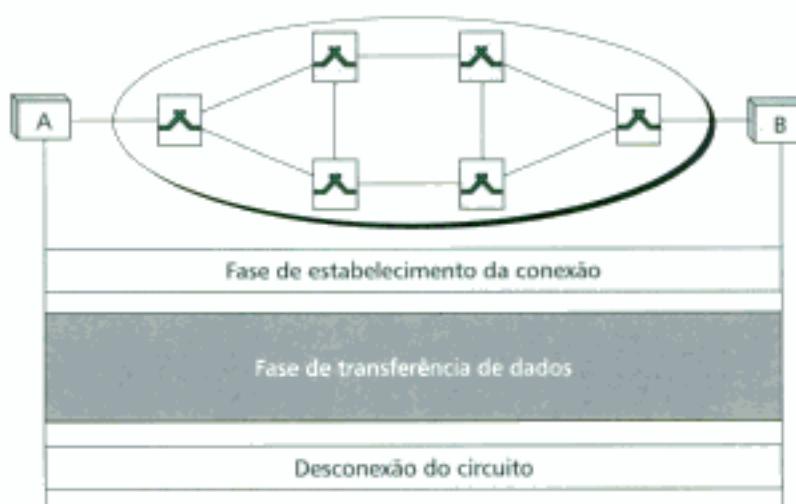


Figura 18.3 Fases VCI.

Fase de Transferência de Dados

Para que seja possível a transferência de *frames* entre a origem e o destino, todos os nós de comutação ao longo da rede precisam ter uma tabela de entradas para este circuito virtual. A tabela, na forma mais simples, possui quatro colunas. Isto significa que cada nó de comutação sustenta quatro partes da informação para cada circuito virtual estabelecido. Adiante mostraremos como os nós conseguem criar tais tabelas. No momento, assumiremos que cada nó possui a respectiva tabela com as entradas ativas para o circuito virtual. A Figura 18.4 ilustra um nó de comutação e a tabela em questão.

A figura mostra um *frame*, cujo VCI = 14, chegando à porta 1 do nó. No comutador, o *frame* é verificado e comparado com os dados da tabela interna do nó para determinar o identificador do *frame* na porta 1, sendo encontrado VCI = 14. Após essa etapa, o nó modifica o VCI para 22 e encaminha o *frame* para a porta 3.

A Figura 18.5 mostra como um *frame* que partiu da origem de dados A alcança o destino B e de que forma o identificador VCI é modificado durante a viagem. Cada nó de comutação modifica o VCI e roteia o *frame*.

A fase de transferência de dados permanece ativa até que a origem consiga transmitir todos os *frames* endereçados ao destino particular. O nó comutação procede da mesma forma para cada *frame* de informação. Esse processo cria um circuito virtual entre a origem de dados e o destino.

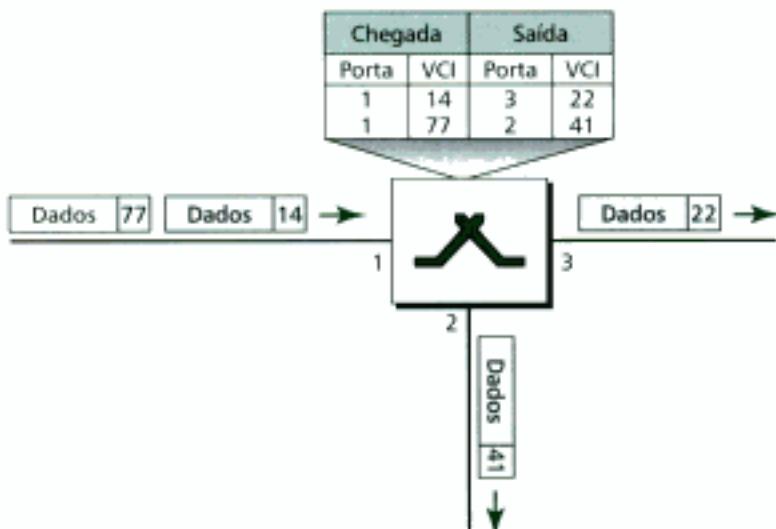


Figura 18.4 Nô de comutação (comutador) e a respectiva tabela de comutação.

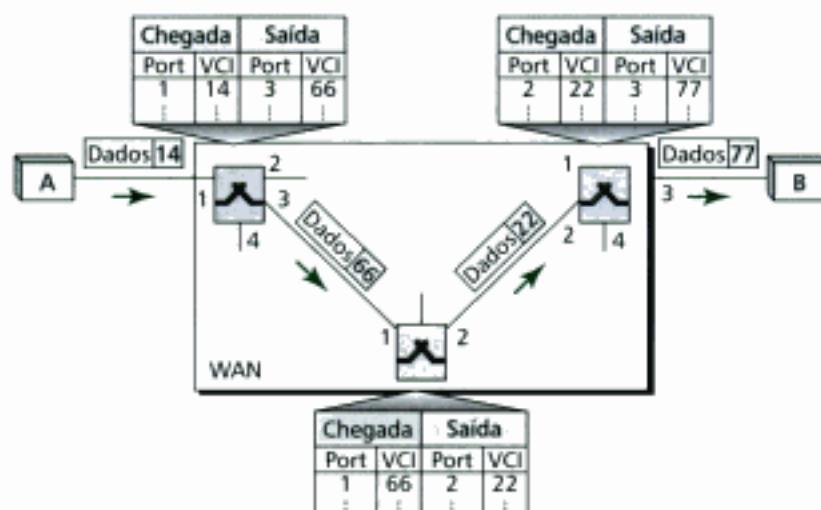


Figura 18.5 Transferência de dados entre origem e destino.

Fase de Estabelecimento da Conexão

A fase de estabelecimento da conexão é bastante interessante. Como um nó de comutação consegue criar uma entrada na tabela para um circuito virtual? Existem duas maneiras de se estabelecer um circuito virtual: **círculo virtual permanente (Permanent Virtual Circuit – PVC)** e o **círculo virtual comutado (Switched Virtual Circuit – SVC)**.

Círculo Virtual Permanente

A origem de dados e o destino podem negociar o estabelecimento do circuito virtual permanente. Neste caso, o estabelecimento da conexão é muito simples. Os circuitos estão sempre disponíveis pois são configurados remota e eletronicamente pelo administrador da rede. O valor do VCI de saída é atribuído à origem e o VCI de chegada é atribuído ao destino. A origem sempre utilizará este VCI para transmitir *frames* ao destino especificado pelo campo de endereçamento. Nesse caso, ao receber os *frames* e olhar o VCI de chegada, o destino saberá que os *frames* vieram da mesma fonte. Se houver necessidade de comunicação *full-duplex* são estabelecidos dois circuitos virtuais. Se houver uma linha telefônica alugada entre A e B, a origem A pode ajustar o receptor dela de modo a manter a comunicação com B sem a necessidade de discagem.

Círculo Virtual Comutado

As conexões PVC possuem duas desvantagens. Primeira, elas têm custo mais elevado porque as duas partes pagam pela conexão durante todo o tempo (até mesmo quando não estiverem se comunicando). Segunda, uma conexão é criada entre uma origem e um único destino.

Uma alternativa interessante é a comutação SVC. A técnica SVC estabelece um circuito temporário entre a origem e o destino. Assim, os sistemas SVC requerem fortemente a fase da conexão. Suponha que a origem A deseja estabelecer um circuito virtual com B. São requeridas duas etapas: (1) solicitação e (2) confirmação do circuito (ACK).

Solicitação do Circuito Um *frame* de solicitação de circuito é transmitido de A até B. A Figura 18.6 ilustra o processo.

1. A origem de dados A transmite um *frame* de solicitação ao nó de comutação I.
2. O nó de comutação I recebe o *frame* de solicitação de A. Ele sabe que um *frame* de A para B tem que passar pela porta 3 (veremos nos capítulos futuros como o nó de comutação obteve esta informação). Na fase de solicitação, o nó funciona como um roteador possuindo uma tabela de roteamento que é, evidentemente, diferente da tabela de comutação. Para o momento, assuma que o nó simplesmente conhece a porta de

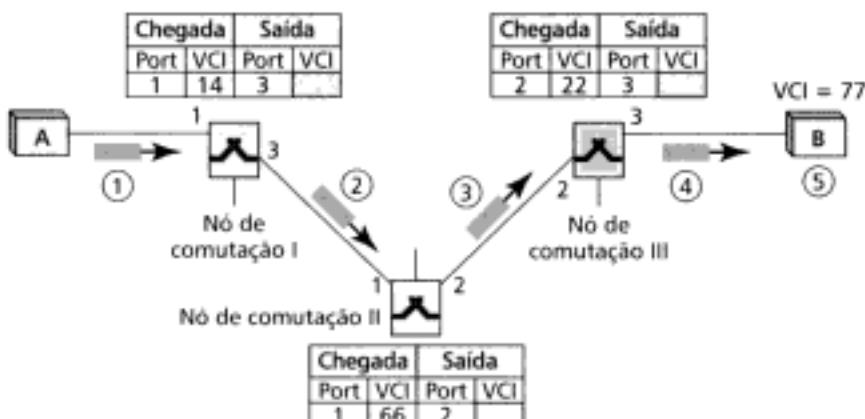


Figura 18.6 Fase de solicitação do SVC.

saída. Assim, é gerada uma entrada na respectiva tabela de comutação identificando este circuito virtual. Entretanto, o nó de comutação preenche apenas três colunas na tabela. O nó atribui à porta de entrada o número 1, escolhe o número VCI de chegada como 14 e identifica a porta de saída como a de número 3. Ele ainda não sabe qual será o número VCI de saída, o que será determinado durante a etapa de confirmação. Então, o nó de comutação I direciona o *frame* para o nó de comutação II, através da porta número 3.

- O nó de comutação II recebe o *frame* de solicitação. Nesse estágio ocorrem os mesmos eventos citados para o nó I. As três colunas da tabela de comutação de II são preenchidas. Assim, a porta de entrada é identificada como a porta 1, o número VCI de chegada é 66 e a porta de saída é a de número 2.
- O nó de comutação III recebe o *frame* de solicitação. Novamente, as três colunas da tabela desse nó são completadas: a porta de entrada é identificada com o número 2, o número VCI de chegada é 22 e a porta de saída é a número 3.
- O destino B recebe o *frame* de solicitação e, se estiver pronto para receber *frames* de A, atribui um número VCI ao *frame* de chegada recebido de A (77). Este número VCI informa ao destino que os *frames* que estão sendo recebidos vieram de A e não de outros endereços.

Confirmação (ACK) Um *frame* especial, denominado *frame* de confirmação ou ACK *frame*, consegue completar as entradas restantes nas tabelas de comutação. A Figura 18.7 mostra o processo.

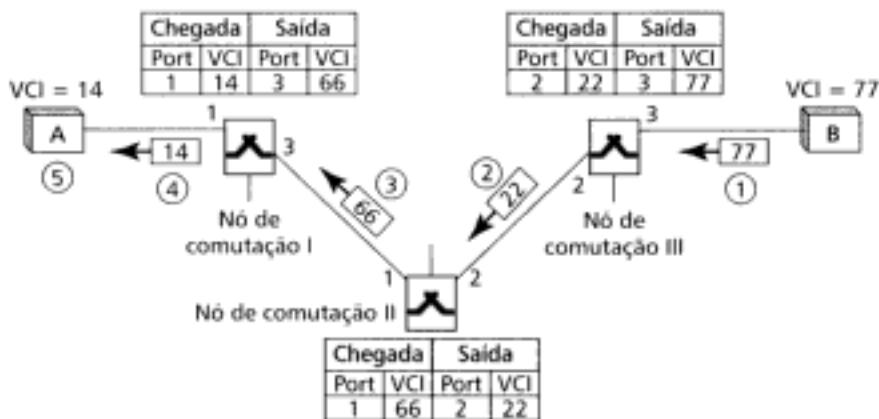


Figura 18.7 Fase de confirmação (ACK) do SVC.

1. O nó de destino transmite um ACK ao nó de comutação III. O ACK carrega os endereços globais de origem e destino de maneira que os nós de comutação aprendam, preenchendo a entrada vazia na tabela. O ACK frame carrega o número VCI 77, escolhido pelo nó de destino como número VCI de chegada dos frames oriundos de A. O nó de comutação III utiliza este número VCI para completar a coluna VCI restante para essa entrada. Perceba que 77 é o número VCI de chegada para o destino B, além de ser o número VCI de saída do nó III.
2. O nó de comutação III retransmite um ACK ao nó II contendo o número VCI de chegada na tabela, escolhido na etapa anterior. O nó II utiliza este número como VCI de saída na tabela.
3. O nó de comutação II retransmite um ACK ao nó I contendo o número VCI de chegada na tabela, escolhido na etapa anterior. O nó I utiliza este número como VCI de saída na tabela.
4. Finalmente, o nó de comutação I transmite um ACK à origem A contendo o número VCI de chegada na tabela, escolhido na etapa anterior.
5. A origem utiliza este número como VCI de saída para os frames de dados a serem transmitidos ao destino B.

Fase de Desconexão do Circuito

Nesta fase, a origem A transmite um frame especial, denominado frame de desconexão, ao destino B. O destino B responde com um frame de confirmação de desconexão. Todos os nós apagam as entradas nas tabelas correspondentes ao circuito virtual que interliga A e B.

18.2 FRAME RELAY

Frame Relay é um circuito virtual WAN desenvolvido no final dos anos 80, início dos anos 90, para responder às demandas de novos tipos de serviços em redes WAN.

1. Antes do Frame Relay, algumas empresas utilizavam uma rede de comutação de circuitos virtuais denominada **X.25** que realizava comutação até o nível da camada de rede. Por exemplo, a Internet utiliza as redes WANs para transportar pacotes de dados de um ponto a outro e isso era feito pelo X.25. Em alguns lugares, o X.25 ainda é utilizado para prover serviços de Internet, mas hoje ele está sendo sistematicamente substituído por outras tecnologias. O problema é que o X.25 possui muitas desvantagens:
 - a. O padrão X.25 tem uma taxa de transmissão baixa (64 kbps). No início dos anos 90 houve uma necessidade de WANs mais velozes e o padrão X.25 não atendia às expectativas.
 - b. O padrão X.25 baseia-se num mecanismo de controle de fluxo e erros muito amplo, abrangendo tanto a camada de enlace quanto a camada de rede. Isto se deve ao fato do X.25 ter sido desenvolvido nos anos 70. Naquela época os meios de transmissão disponíveis eram bastante susceptíveis a erros. O problema é que fazer controle de fluxo e erro em ambas camadas cria uma quantidade muito grande de *overhead*, reduzindo sensivelmente as taxas de transmissão. Além disso, o X.25 requer ACKs para os frames da camada de enlace e para os pacotes da camada de rede, transmitidos entre a origem e o destino.
 - c. Originalmente, o X.25 foi desenvolvido para uso privado e não para a Internet. Assim, o X.25 tinha sua própria camada de rede. Isto significa que os dados dos usuários são encapsulados em pacotes X.25 na camada de rede. Entretanto, a Internet tem uma camada de rede própria. Assim, se quisermos utilizar o X.25 como meio de obtenção de acesso à Internet, os pacotes da camada de rede do modelo da Internet, denominados datagramas, devem ser entregues ao protocolo X.25 para encapsulamento no pacote X.25. Isto duplica a geração de *overhead*.

2. Desapontadas com o X.25, algumas empresas iniciaram a construção de redes WANs privadas, partindo do aluguel das linhas T-1 ou T-3 (tributários) das operadoras de telefonia pública. Esta escolha também teve algumas desvantagens:
 - a. Se a empresa tivesse n filiais espalhadas numa determinada área geográfica, ela necessitava $n(n - 1)/2$ linhas T-1 ou T-3. A empresa pagava por todas essas linhas, embora as utilizava apenas 10% do tempo. É claro que isso era muito dispendioso.
 - b. Os serviços oferecidos pelas linhas T-1 e T-3 assumem que o usuário possui uma taxa de transmissão de dados fixa. Por exemplo, uma linha T-1 foi desenvolvida para serviços que utilizam a linha continuamente a 1,544 Mbps. Hoje em dia, isto não é conveniente a muitos serviços que necessitam transmitir **rajadas de dados**. Por exemplo, um determinado serviço pode querer transmissão de dados a 6 Mbps durante 2 s, 0 Mbps (sem transmissão) durante 7 s e 3,44 Mbps durante 1 s, perfazendo um total de 15,44 Mbit/s durante 10 s. Embora a taxa de transmissão média ainda seja 1,544 Mbps, a linha T-1 não pode aceitar este tipo de demanda porque ela foi desenvolvida para oferecer taxas de transmissão fixas, não rajadas de dados. As rajadas requerem o que é denominado **banda sob demanda**. Nesses serviços, os usuários necessitam de diferentes larguras de bandas em instantes de tempo diferentes.

O protocolo Frame Relay foi criado de modo a cobrir o buraco deixado pelo protocolo X.25. O Frame Relay é um protocolo de rede WAN com as seguintes características:

1. O Frame Relay opera em velocidades muitas altas (o mais antigo a 1,544 Mbps e, o mais recente, a 44,376 Mbps). Isto significa que esse sistema pode facilmente ser utilizado no lugar das linhas T-1 ou T-3.
2. O Frame Relay opera apenas nas camadas física e de enlace. Esta característica o torna especialmente adequado nas redes *backbone* oferecendo serviços às aplicações que realmente têm um protocolo de camada de rede, tal como a Internet.
3. O Frame Relay permite rajada de dados.
4. O Frame Relay permite um *frame* de tamanho até 9000 bytes. Este *frame* pode acomodar todos os *frames* utilizados nas redes LANs.
5. O serviço de Frame Relay é mais barato que os demais serviços de WANs tradicionais.
6. O Frame Relay detecta erros somente na camada de enlace. Não existe mecanismos de controle de fluxo ou de erros. Não existe nem mesmo uma política de retransmissão de *frames* danificados. O Frame Relay foi projetado de modo a proporcionar altas taxas de transmissão através de meios confiáveis, principalmente para os protocolos de camadas mais altas que dispõem de mecanismos de controle de fluxo e de erros.

Arquitetura

A tecnologia Frame Relay provê comunicação tanto em circuitos virtuais permanentes (PVC) quanto em circuitos virtuais comutados (SVC). A Figura 18.8 mostra um exemplo de rede Frame Relay conectada à Internet. Como veremos no Capítulo 19, os roteadores são utilizados para conectar as LANs e WANs na Internet. Na figura, a rede WAN Frame Relay é utilizada como *link* de acesso à Internet global.

Circuitos Virtuais

A rede Frame Relay é uma rede de circuitos virtuais. Um circuito virtual na rede Frame Relay é identificado por um número denominado **DLCI (Data Link Connection Identifier)**. O Frame Relay usa tanto os PVCs quanto os SVCs.

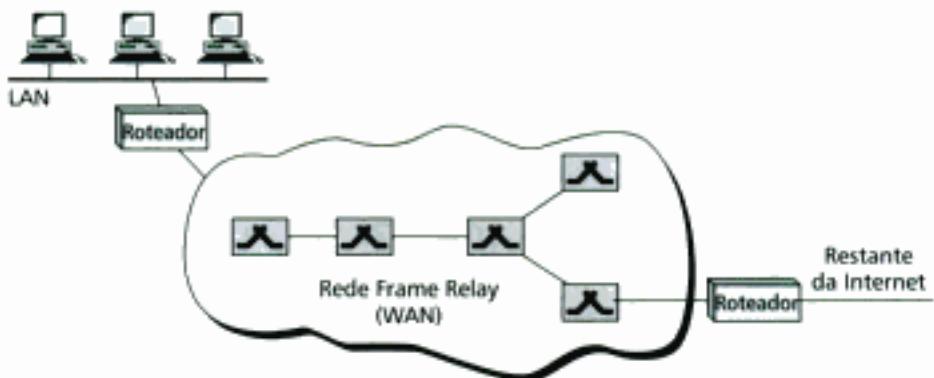


Figura 18.8 Rede Frame Relay.

Nas redes Frame Relay, os VCIs são denominados DLCI.

Nós de Comutação

Cada nó de comutação numa rede Frame Relay possui uma tabela para rotear *frames*. A tabela associa a combinação porta de chegada-DLCI a uma combinação porta de saída-DLCI, como descrevemos para a teoria geral da comutação de circuitos virtuais. A única diferença é que os números VCIs devem ser substituídos pelos DLCIs.

Camadas Frame Relay

A Figura 18.9 ilustra o modelo de camadas Frame Relay. O Frame Relay possui apenas as camadas física e de enlace de dados.

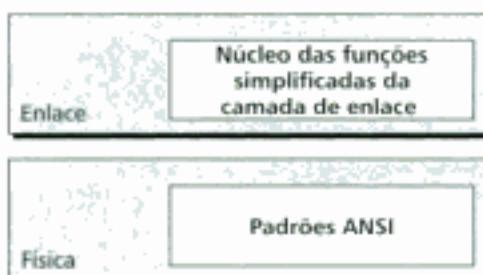


Figura 18.9 Camadas Frame Relay.

O Frame Relay opera somente nas camadas física e de enlace de dados.

Camada Física

A camada física Frame Relay não define nenhum protocolo específico. Em vez disso, deixa o desenvolvedor usar tudo o que estiver disponível nela. Por isso, o Frame Relay suporta qualquer protocolo reconhecido pela ANSI.

Camada de Enlace de Dados

Na camada de enlace, o Frame Relay emprega uma versão simplificada do HDLC. É utilizada esta versão simplificada porque o HDLC oferece recursos de controle de fluxo e de erros bastante extensos e que não são necessários no protocolo Frame Relay.

A Figura 18.10 mostra o formato característico do *frame* na tecnologia Frame Relay. A estrutura do *frame* é bastante semelhante ao HDLC. De fato, os campos *flag*, FCS e informação são os mesmos. Entretanto, o campo controle foi descartado porque esse campo é dedicado ao controle de fluxo e de erros no HDLC, recursos desnecessários no Frame Relay. O campo de endereços define o número DLCI, assim como alguns dos bits utilizados para controle de congestionamento e de tráfego.

C/R: Comando/resposta
 EA: Endereço estendido
 FECN: Encaminhar notificação de congestionamento explícito

BECN: Notificação inversa de congestionamento explícito
 DE: Tráfego marcado
 DLCI: Campo endereço

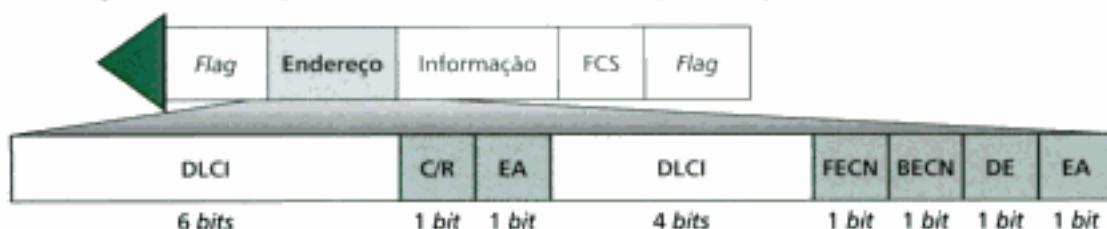


Figura 18.10 Formato do frame Frame Relay.

As descrições dos campos são as seguintes:

- **Campo Endereço (DLCI).** Os primeiros 6-bits são dedicados à parte 1 do identificador DLCI. A segunda parte do DLCI usa os quatro primeiros bits do segundo byte. Estes bits fazem parte do identificador de enlace e somados produzem os 10-bits definidos pelo padrão. A função do DLCI foi discutida anteriormente. Veremos a questão do endereçamento estendido no final desta seção.
- **Bit Comando/Resposta (Command/Response – C/R).** O bit C/R (comando/resposta) permite às camadas mais altas identificarem o frame como um comando ou uma resposta. Esse campo geralmente não é utilizado pelo protocolo Frame Relay.
- **Bit de Endereço Estendido (Extended Address – EA).** O bit EA indica se o byte atual tem o endereço final. O bit EA = 0 indica que deve ser esperado outro byte de endereço. Se EA = 1 significa que o byte atual contém o endereço final.
- **Bit FECN (Forward Explicit Congestion Notification).** Este bit é configurado por qualquer nó de comutação para indicar que o tráfego está congestionado na direção de viagem do frame. Esse bit informa ao nó de destino dos dados que ocorreu congestionamento do circuito. No Capítulo 23 discutiremos melhor o uso desse bit quando estivermos tratando controle de congestionamento.
- **Bit BECN (Backward Explicit Congestion Notification).** O bit BECN é configurado de modo a indicar a existência de congestionamento na direção oposta à direção que o frame está viajando. Esse bit informa o congestionamento ao nó de origem. Outra vez, focaremos o uso desse bit no Capítulo 23.
- **Bit DE (Discard Eligibility).** O bit DE indica o nível de prioridade do frame. Nas situações de emergência, os nós de comutação podem ser obrigados a descartar frames para aliviar gargalos e manter a rede livre dos colapsos provocados pela sobrecarga de informação. Quando DE = 1, esse bit informa aos nós da rede que o frame deve ser descartado caso haja congestionamento. Este bit pode ser configurado tanto pela origem de dados como pelos nós de comutação ao longo do percurso. Voltaremos a esse bit quando estivermos analisando as questões relativas ao controle de congestionamento no Capítulo 23.

O protocolo Frame Relay não fornece controle de fluxo e erros. Esses recursos devem ser fornecidos pelos protocolos das camadas superiores.

Endereço Estendido

Para aumentar a faixa de DLCIs, o endereço Frame Relay foi estendido de 2 para 3 ou 4-bytes. A Figura 18.11 mostra três endereços DLCI diferentes. Perceba que o campo EA define a quantidade de bytes. Ele vale 1 no último byte de endereço e 0 nos demais bytes. Note ainda que nos formatos de 3 ou 4-bits o penúltimo bit é configurado para 0.

DLCI			C/R	EA=0
DLCI	FECN	BECN	DE	EA=1

a. Endereço de 2 bytes (DLCI de 10-bits)

DLCI			C/R	EA=0
DLCI	FECN	BECN	DE	EA=1
DLCI			0	EA=1

b. Endereço de 3 bytes (DLCI de 16-bits)

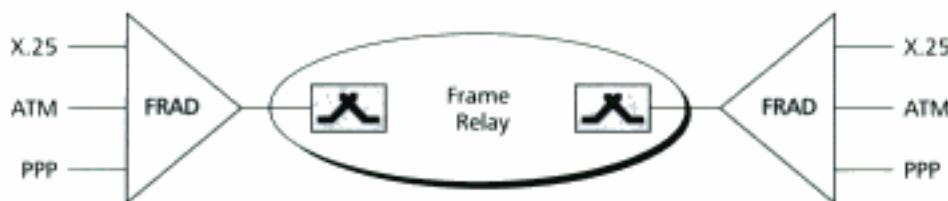
DLCI			C/R	EA=0
DLCI	FECN	BECN	DE	EA=0
DLCI			EA=0	EA=0
DLCI			0	EA=1

c. Endereço de 4 bytes (DLCI de 23-bits)

Figura 18.11 Três formatos DLCIs diferentes.

FRADs

Para tratar *frames* recebidos de outros protocolos o Frame Relay usa um dispositivo denominado **FRAD** (**Frame Relay Assembler/Disassembler**). Um FRAD monta e desmonta *frames* recebidos de outros protocolos para permitir que eles sejam transportados pelos *frames* padrão Frame Relay.

**Figura 18.12** FRAD.

Um FRAD pode ser um dispositivo à parte dos nós de comutação ou podem ser incorporados neles. A Figura 8.12 ilustra dois FRADs conectando protocolos diferentes a uma rede Frame Relay.

VOFR

As redes Frame Relay oferecem uma opção de serviço denominada **Voz Sobre Frame Relay (Voice Over Frame Relay – VOFR)** para transmitir voz através da rede. O sinal de voz é digitalizado num sistema PCM e comprimido logo em seguida. O resultado é transmitido como *frame* de dados através da rede. Esta característica possibilita transmissões de voz a baixo custo através de longas distâncias. Entretanto, perceba que é inevitável a perda de qualidade do sinal de voz recuperado. De fato, o sinal recuperado no VOFR está longe de ter a qualidade dos sinais transmitidos nas redes de telefonia públicas (RTPCs).

LMI

O protocolo Frame Relay foi projetado originalmente para fornecer conexões PVC. Desse modo, não existiam provisões de interfaces de gerenciamento e controle no Frame Relay. O protocolo LMI (**Local Management Information**) foi incorporado recentemente ao protocolo Frame Relay de modo a tratar as questões relativas ao gerenciamento. Em particular, o LMI pode prover:

- Um mecanismo de *keep-alive* para verificar se os dados estão fluindo na rede.
- Um mecanismo *multicast* permitindo muitos sistemas locais transmitir *frames* a muitos sistemas remotos.
- Um mecanismo para permitir que os sistemas finais verifiquem o *status* dos nós de comutação (por exemplo, para verificar se o nó está congestionado).

Controle de Congestionamento e Qualidade de Serviços

Uma excelente característica do Frame Relay é o suporte ao **controle de congestionamento** e **qualidade de serviços**. Não abordamos este tipo de questão ainda. No Capítulo 23, introduziremos estes aspectos importantes das redes e discutiremos como eles são implementados nas redes Frame Relay, TCP/IP e outras.

18.3 ATM

ATM (Asynchronous Transfer Mode) é um protocolo de **comutação de células** desenvolvido pelo fórum ATM e adotado pelo ITU-T. O ATM é designado para aproveitar os meios de transmissão de alta velocidade, como, por exemplo, E3, SONET e T3. De fato, as redes ATM podem ser pensadas como a "highway" da informação.

Metas do Projeto ATM

Destacamos seis desafios dentre os enfrentados pelos projetistas do ATM:

1. Em primeiro lugar era necessário um sistema de transmissão de dados que otimizasse a utilização dos meios de transmissão banda larga, como as fibras ópticas. Além disso, para oferecer banda larga efetiva, os meios de transmissão e os equipamentos de rede deveriam ser drasticamente menos susceptíveis à degradação por ruídos. Era necessária uma tecnologia que tirasse vantagem de ambos fatores e, por meio disso, maximizasse as taxas de transmissão.
2. O sistema deveria ser capaz de interfacear com os sistemas existentes na época e permitir conexão WAN entre eles sem perda de efetividade da rede ou requer a substituição desses sistemas.
3. O projeto deveria ter um custo de implementação reduzido para que não fossem colocadas barreiras à adoção do padrão (não em termos de custos). Se o padrão ATM tornasse de fato o controlador principal do *backbone* de comunicações internacional, como era pretendido, ele deveria estar disponível a baixo custo para qualquer usuário que desejasse utilizá-lo.
4. O novo sistema deveria ser capaz de trabalhar com e suportar as hierarquias de telecomunicações existentes (malhas locais, operadoras locais, operadoras de longa distância e assim por diante).
5. O novo sistema deveria ser orientado à conexão para garantir entregas de células ou pacotes.
6. Por último, mas não menos importante, as células de comprimento prefixado deveriam permitir, tanto quanto possível, que o processamento delas ocorresse no *hardware*, reduzindo assim, os atrasos no trânsito provocados pelas funções de processamento de *software*.

Problemas

Antes de discutirmos as soluções adotadas para as metas de projeto é bastante instrutivo examinarmos alguns dos problemas associados às redes existentes na época do surgimento do ATM.

Redes de Pacotes

Antes do ATM, a comunicação de dados no nível de enlace era baseada na comutação ou em redes de pacotes. Diferentes tipos de protocolos usavam *frames* complexos e de tamanhos variáveis. dessa forma, o gerenciamento das redes estava ficando complexo, aumentando consequentemente, a complexidade do transporte da informação útil. Assim, os cabeçalhos dos *frames* estavam ficando muito grandes, quando comparados aos tamanhos das unidades de dados (*payload*). Em resposta

ao aumento do cabeçalho, alguns protocolos também tiveram o tamanho do campo de dados modificado, especialmente para tornar o cabeçalho e, por extensão, todo o *frame* mais eficiente. Infelizmente, campos de dados grandes provocavam desperdício de banda. Se não houvesse muita informação a ser transmitida pelo *frame*, a maior parte do campo de dados não era preenchida com informação útil. Para melhorar a *performance* do pacote, alguns protocolos forneceram pacotes de tamanhos variáveis aos usuários, mas isso mostrou-se uma solução paliativa.

Redes com Tráfego Misto

Como você pode imaginar, a variedade de *frames* tornou o tráfego imprevisível. Os *switches*, multiplexadores e roteadores eram obrigados a incorporar *softwares* (sistemas operacionais) de modo a gerenciar os diversos tipos e tamanhos de *frames* (aliás, problema que ainda persiste). A grande quantidade de informação nos cabeçalhos afetava diretamente a *performance* das redes, pois cada *bit* do *frame* deveria ser verificado e avaliado para garantir a integridade de cada *frame*. O processo de *internetworking* dos diferentes tipos de *frames* entre as redes era lento, caro e/ou impossível (na pior das hipóteses).

Outro problema era garantir a entrega de *frames* a uma taxa de dados consistente, já que os tamanhos dos *frames* são imprevisíveis. Para tirar o melhor que a tecnologia banda larga tem a oferecer, o tráfego teve que ser multiplexado no tempo (TDM) para dentro do meio compartilhado. Imagine os resultados da multiplexação de *frames* entre duas redes com requerimentos e tipos de *frames* diferentes, compartilhando um único *link* (veja a Figura 18.13). O que acontece quando a linha 1 usa *frames* grandes (usualmente *frames* de dados) enquanto a linha 2 utiliza *frames* muito pequenos (padrão para informação de áudio e vídeo)?

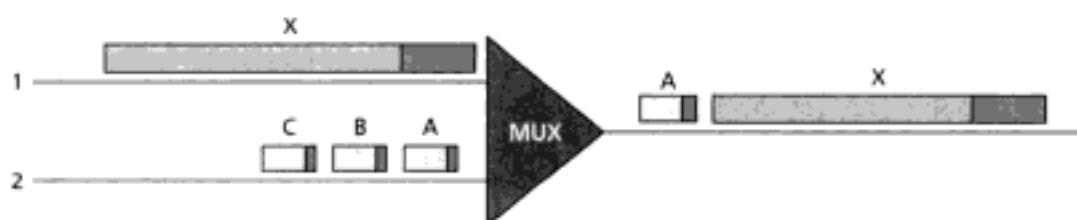


Figura 18.13 Multiplexação de *frames* de tamanhos diferentes.

Se o gigantesco *frame* X da linha 1 chegar ao multiplexador ligeiramente à frente dos *frames* da linha 2, o multiplexador não terá outra escolha senão colocar o *frame* X na direção do canal de saída. O multiplexador não tem como saber quanto tempo esperar para que todo o *frame* X seja alocado na saída, até mesmo se os *frames* da linha 2 tiverem prioridade mais alta sobre o *frame* da linha 1. Logo, o *frame* A deve esperar até que todo o *frame* X seja movido na direção da saída. O redirecionamento do *frame* X cria um atraso indesejável do *frame* A. O mesmo desequilíbrio pode afetar todos os outros *frames* da linha 2.

Já que os *frames* de áudio e vídeo são tradicionalmente pequenos, misturá-los com o fluxo de dados convencional cria freqüentemente atrasos inaceitáveis, tornando o compartilhamento dos *links* pouco útil para esse tipo de informação. O tráfego desses dois tipos de dados deveria viajar por diferentes caminhos, como acontece com o tráfego de carros e trens numa cidade ou região. Mas para utilizar toda a largura de banda dos *links* devemos ser capazes de transmitir quaisquer tipos de tráfego através dos mesmos *links*! Vejamos que solução as redes ATM apresentam para esse dilema.

Redes de Células

Muitos dos problemas associados à *internetworking* de *frames* são resolvidos via adoção de um conceito denominado rede de células. Uma célula é uma pequena unidade de dados de tamanho fixo. Numa **rede de células**, a qual utiliza a **célula** como unidade de dados, todos os dados são carregados em células uniformes capazes de serem transmitidas com total previsibilidade. Quando *frames* de diferentes tamanhos e formatos alcançam a rede de células de uma rede tributária, eles são divididos em pequenas unidades de dados de igual tamanho e são carregados nas células. Tais

células são então multiplexadas juntamente com as demais e roteadas através da rede de células. Visto que cada célula tem o mesmo tamanho e todas são muito pequenas, os problemas associados à multiplexação de frames de tamanhos diferentes são evitados.

Uma rede de células usa a célula como unidade básica de dados. Uma célula é definida como um pequeno pacote de informação de tamanho fixo.

A Figura 18.14 ilustra o mesmo multiplexador da Figura 18.13 com as duas linhas transmitindo células em vez de frames. O frame X foi fragmentado em três células: X, Y e Z. Uma única célula da linha 1 é comutada para a linha de saída antes que a primeira célula da linha 2 tome o lugar dela. As células das duas linhas são separadas de modo tal que nenhuma delas sofre um atraso considerável.

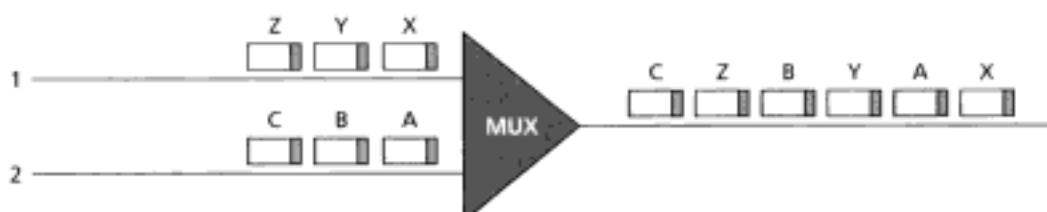


Figura 18.14 Multiplexação de células.

O segundo ponto nesse mesmo cenário é que a elevada taxa de transmissão de dados desses links, aliada ao tamanho pequeno das células, sugere que as células de cada linha chegam aos respectivos destinos formando, aproximadamente, um fluxo contínuo de informação (desconsiderando as separações intercelulares). A situação é análoga ao que acontece quando você assiste a um filme. As seqüências em sua mente parecem ser contínuas, mas de fato o filme é formado por seqüências de quadros justapostos, apresentados de modo a provocar ilusão de continuidade das cenas. Desse modo, uma rede de células pode controlar transmissões em tempo real, tal como numa chamada telefônica, sem que as partes tomem consciência de que estão sendo fragmentados e multiplexados.

TDM Assíncrono

ATM utiliza o conceito de multiplexação por divisão de tempo para multiplexar as células oriundas de diferentes canais. Por essa razão é denominada Asynchronous Transfer Mode (modo de transferência assíncrono). Ela utiliza slot-times de tamanhos fixos (tamanho de uma célula). Os multiplexadores ATM preenchem cada slot-times com uma célula do canal de entrada selecionado e que tiver células para transmitir. O slot só é deixado vazio se nenhum dos canais de entrada tiver células para transmitir.

A Figura 18.15 ilustra como células de três canais de entrada são multiplexadas. No primeiro pulso de relógio (clock), o canal 2 não tem célula para transmitir (slot de entrada vazio). Desse modo, o multiplexador preenche o slot com uma célula do terceiro canal. Quando todas as células de todos os canais tiverem sido multiplexadas, os slots de saída ficam vazios.



Figura 18.15 Multiplexação ATM.

Arquitetura

ATM é uma rede de comutação de células. O usuário utiliza dispositivos finais, denominados **User-to-Network Interface (UNI)**, para se conectar aos *switches* dentro da rede de células. Os *switches* são conectados mutuamente através das **Network-to-Network Interfaces (NNIs)**. A Figura 18.16 mostra um exemplo de rede ATM.

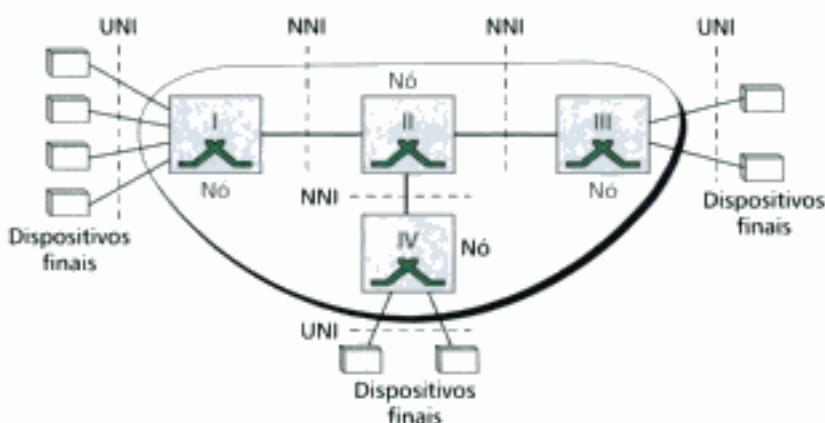


Figura 18.16 Arquitetura de uma rede ATM.

Conexão Virtual

A conexão entre dois dispositivos finais é realizada através dos caminhos de transmissão (TPs), caminhos virtuais (VPs) e circuitos virtuais (VCs). Um **caminho de transmissão** é uma conexão física (fio, cabo, satélite e assim por diante) entre um UNI e um nó de comutação ou entre dois nós de comutação. Pense em dois nós de comutação como duas cidades. O caminho de transmissão é o conjunto de todos meios de acesso banda larga que conecta diretamente as duas cidades.

Um caminho de transmissão é dividido em muitas rotas virtuais. Um **caminho virtual (VP)** proporciona uma conexão ou um conjunto de conexões entre dois nós de comutação. Imagine que um caminho virtual é uma *highway* de dados conectando duas cidades. Cada *highway* forma um caminho virtual. O conjunto de todas as *highways* é o caminho de transmissão.

As redes de células estão baseadas em **circuitos virtuais (VCs)**. Todas as células pertencentes ao mesmo bloco de informação seguem o mesmo circuito virtual, permanecendo na ordem original até que as células cheguem ao destino. Imagine um circuito virtual como linhas de uma *highway* (caminho virtual). A Figura 18.17 ilustra o relacionamento entre o caminho de transmissão (caminho físico), caminhos virtuais (uma combinação de circuitos virtuais que são agrupados juntos porque as partes dos caminhos deles são idênticos) e circuitos virtuais que conectam logicamente dois pontos.

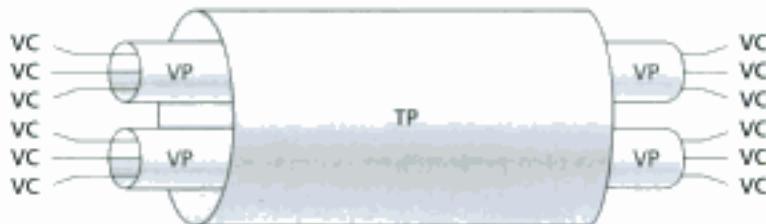


Figura 18.17 TP, VPs e VCs.

Para melhor compreender o conceito de VPs e VCs observe a Figura 18.18. Nesta figura, oito dispositivos finais estão se comunicando através de quatro VCs. Contudo, os dois primeiros VCs parecem compartilhar o mesmo caminho virtual entre os *switches* I e II. Assim, é razoável agrupar esses dois VCs juntos para formar um VP. De outro modo, parece claro que os outros dois VCs compartilham o mesmo caminho entre os *switches* I e IV. Logo, também é razoável combiná-los para formar um VP.

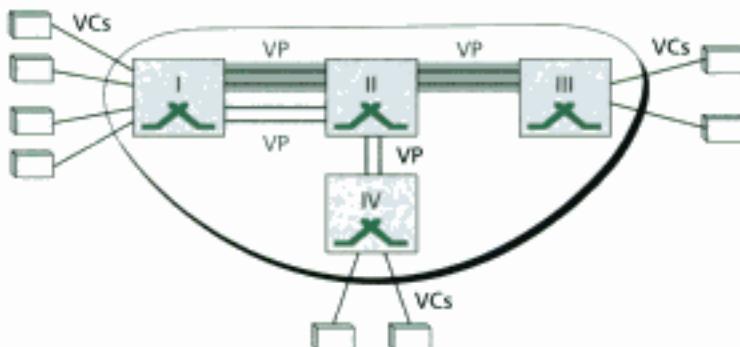


Figura 18.18 Exemplo de VPs e VCs.

Identificadores

Para rotear dados entre dispositivos finais de um circuito virtual, as conexões virtuais precisam ser devidamente identificadas. Para esta finalidade, os projetistas do ATM criaram um identificador hierárquico composto de dois níveis: **VPI (Virtual Path Identifier)** e **VCI (Virtual Circuit Identifier)**. O VPI define um VP específico e o VCI define um VC particular dentro do VP. O VPI é único para todas as conexões virtuais agrupadas logicamente dentro de um VP.

Toda conexão virtual é identificada pelo par de números VPI e VCI.

A Figura 18.19 mostra os VPIs e VCIs para um caminho de transmissão específico. A razão lógica para dividir um identificador em duas partes quando estivermos discutindo o roteamento numa rede ATM.

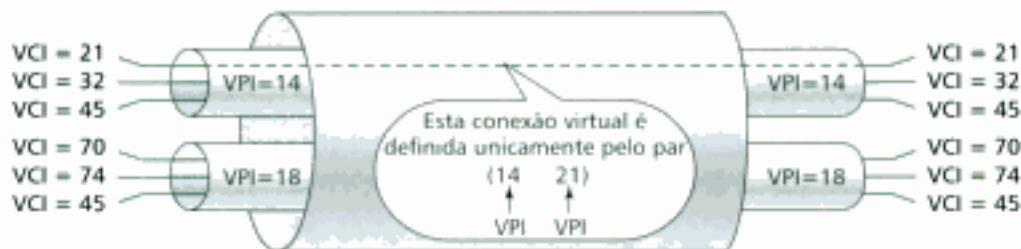


Figura 18.19 Identificadores de conexão.

Os tamanhos dos VPIs para UNIs e NNIs são diferentes. Num UNI, o VPI tem 8-bits, enquanto que num NNI os VPI tem 12-bits. O tamanho do VCI é o mesmo em ambas interfaces. Desse modo, podemos dizer que uma conexão virtual é identificada por 24-bits num UNI e 28-bits num NNI (veja Figura 18.20).

A idéia central por trás da divisão de um identificador de conexão virtual em duas partes é permitir o roteamento hierárquico. A maioria dos nós de comutação numa rede ATM típica são roteados usando os VPIs. Os nós de comutação da borda da rede, isto é, aqueles que interagem diretamente com os dispositivos finais, usam tanto VPIs quanto VCIs.

Células

A unidade básica de dados numa rede ATM é a célula. Uma célula ATM tem somente 53 bytes de tamanho, sendo 5 bytes destinados ao cabeçalho (*header*) e 48 bytes para transporte de dados (*payload*). É importante mencionar que a quantidade de dados transmitidos pelo usuário pode ser inferior a 48 bytes. Estudaremos em detalhes os campos da célula ATM, mas por enquanto é suficiente dizer que a maior parte do cabeçalho é ocupada pelos identificadores VPI e VCI para definir a conexão virtual através da qual a célula ATM seguirá, passando entre os diversos nós de comutação da rede. A Figura 18.21 ilustra a estrutura da célula.

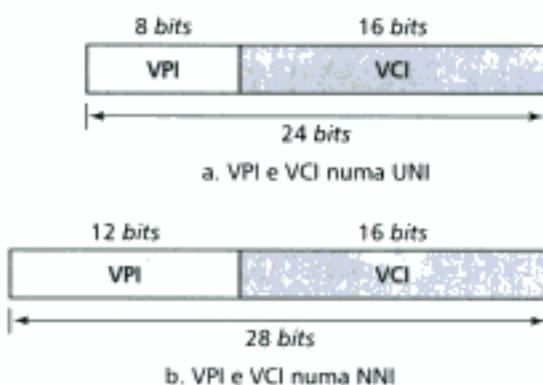


Figura 18.20 Identificadores de conexão virtual em UNIs e NNI.

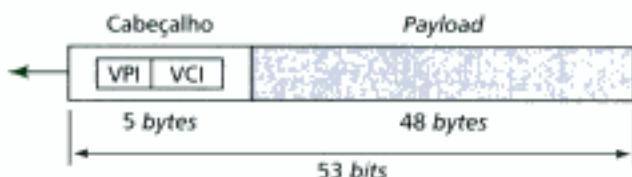


Figura 18.21 Uma célula ATM.

Estabelecimento e Liberação da Conexão

Da mesma forma que o Frame Relay, as redes ATM utilizam dois tipos de conexão: PVC e SVC.

PVC Um circuito virtual permanente (Permanent Virtual Circuit – PVC) é estabelecido entre dois dispositivos finais pelo provedor da rede. Os VPIs e VCIs são definidos para as conexões permanentes e os valores são armazenados nas tabelas de comutação dos nós.

SVC Numa conexão virtual comutada, toda vez que dispositivos finais quiserem fechar uma conexão para troca de dados deve ser estabelecido um novo circuito virtual. O protocolo ATM não consegue realizar o trabalho sozinho, por isso necessita de endereços da camada de rede e serviços de outro protocolo (tal como o IP). O mecanismo de sinalização deste outro protocolo solicita uma requisição de conexão utilizando os endereços da camada de rede dos dois dispositivos finais. O mecanismo real depende do protocolo da camada de rede.

Comutação

O protocolo ATM utiliza comutação para rotear a célula do dispositivo de origem até o destino. Um nó de comutação roteia a célula utilizando tanto os VPIs quanto os VCIs. O roteamento requer todo o mecanismo identificador. A Figura 18.22 mostra como um PVC roteia a célula. Uma célula com um VPI = 153 e VCI = 67 chega ao nó de comutação através da interface 1 (porta 1). O nó verifica a tabela de comutação onde são armazenadas seis partes da informação por linha: número da interface de chegada, VPI de chegada, VCI de chegada, número da interface correspondente de saída, o novo VPI e a nova VCI. O nó determina a entrada com a interface 1, VPI = 153 e VCI = 67 e descobre que a combinação corresponde à interface de saída 3, VPI = 140 e VCI = 92. Então, o nó modifica no cabeçalho o VPI e o VCI para 140 e 92, respectivamente, e retransmite a célula através da interface 3 dele.

Estrutura da Comutação

A tecnologia de comutação introduziu muitas características interessantes com o objetivo de aumentar a velocidade de comutação para controlar os dados. Visto que os nós são utilizados tanto na camada de enlace quanto na camada de rede, não trataremos estes detalhes aqui. Para mais informações, consulte o Apêndice F.

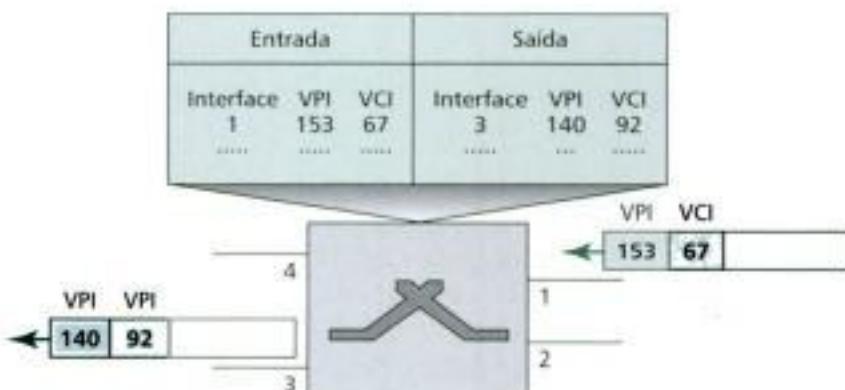


Figura 18.22 Roteamento em um comutador.

Camadas ATM

O padrão ATM define três camadas: camada de adaptação ATM, a camada ATM e a camada física (veja a Figura 18.23).

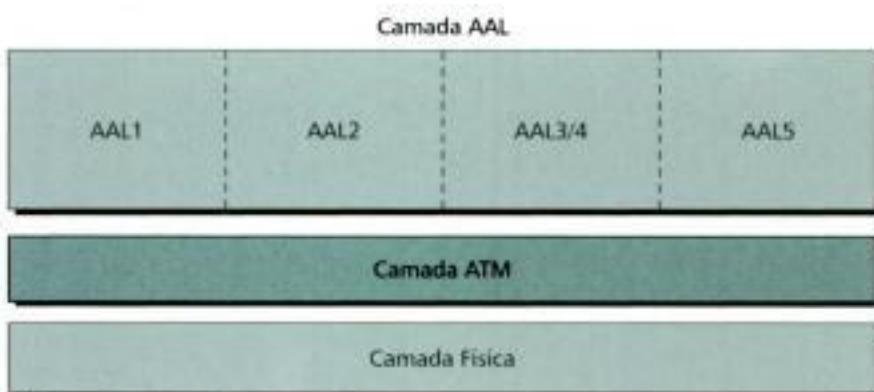


Figura 18.23 Camadas ATM.

Os dispositivos finais utilizam todas as três camadas enquanto os nós de comutação utilizam somente as duas camadas inferiores (veja Figura 18.24).

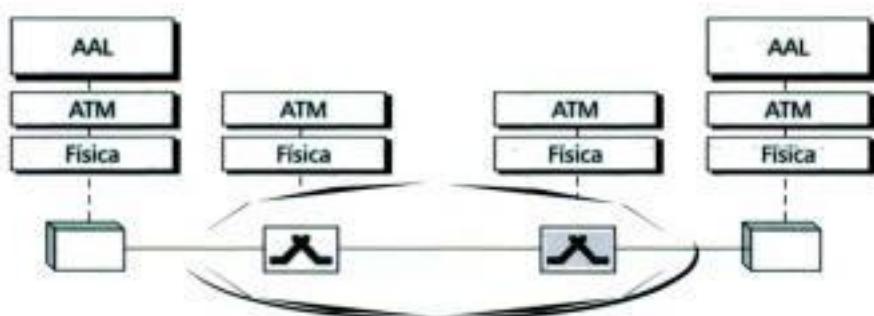


Figura 18.24 As camadas ATM: dispositivos finais e nós de comutação.

Camada Física

Semelhantemente às redes Ethernet e WLANs, as células ATM podem ser transportadas por qualquer meio físico.

SONET O projeto original das redes ATM baseava-se na rede SONET (veja Capítulo 9) como meio físico de transporte. A SONET era preferida por duas razões. Primeiro, as altas taxas de

transmissão de dados da SONET refletiam o projeto e a filosofia ATM. Segundo, usando a SONET as fronteiras das células (início e fim) ficam muito bem definidas. De acordo com o Capítulo 9, a SONET especifica o uso de um ponteiro para mapear o início do *payload*. Se o início da primeira célula ATM estiver definido, o restante das células no mesmo *payload* pode ser facilmente identificado porque não existe intervalos (*gaps*) entre células. Basta contar 53 bytes à frente para determinar a próxima célula.

Outras Tecnologias de Transmissão para ATM O padrão ATM não ficou limitado ao uso da camada física da rede SONET. Outras tecnologias, até mesmo sem fios, já foram utilizadas. Porém, o problema da fronteira da célula deve ser resolvido. Uma solução é o receptor tentar adivinhar o fim da célula e aplicar a técnica do CRC para o cabeçalho de 5 bytes. Se não houver erros, o fim da célula foi determinado corretamente (isto é, com uma probabilidade alta). Basta contar 52 bytes para trás e determinar o início da célula.

Camada ATM

A **camada ATM** oferece serviços de roteamento, gerenciamento de tráfego, comutação e multiplexação. Ela processa todo o tráfego de saída recebendo 48 bytes de segmentos das subcamadas AAL e transformando-os em células de 53 bytes através da adição de um cabeçalho de 5-bytes (veja Figura 18.25).

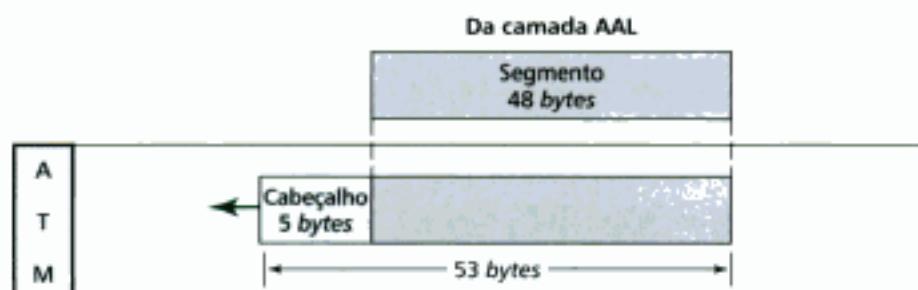


Figura 18.25 Camada ATM.

Formato do Cabeçalho O protocolo ATM usa dois formatos para o cabeçalho, uma para as células UNIs e outro para as células NNIs. A Figura 18.26 ilustra como esses cabeçalhos são organizados num formato definido pelo ITU-T (cada linha representa um byte).

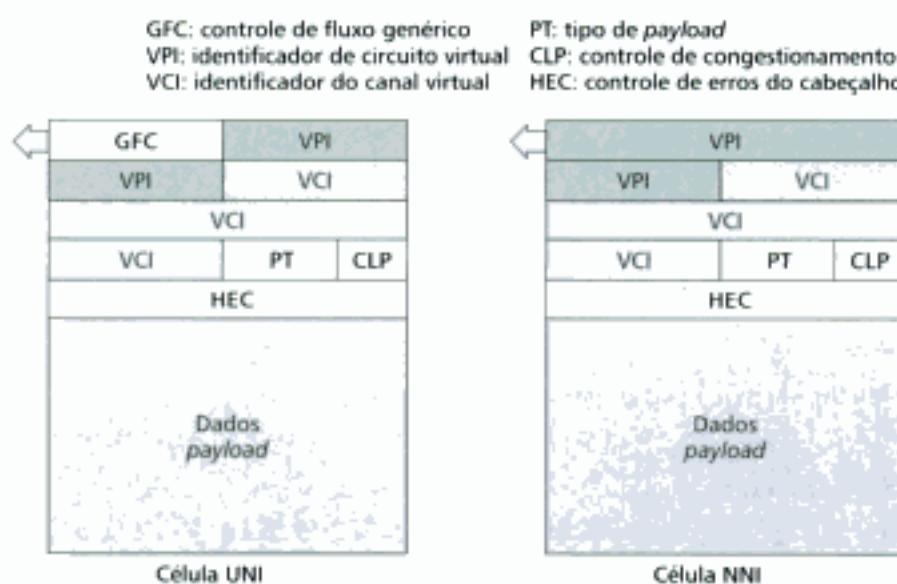


Figura 18.26 Cabeçalhos ATM.

- **Controle de fluxo genérico (Generic Flow Control – GFC).** O campo de 4-bits GFC provê controle de fluxo no nível UNI. O ITU-T determinou que este nível fique restrito ao cabeçalho das células UNI. Assim, no cabeçalho NNI estes bits são adicionados ao VPI. Este VPI estendido permite endereçar mais caminhos virtuais. Algumas alternativas para uso deste campo seriam para marcar como ociosa a célula, ou para marcá-la como sendo de informação de manutenção e operação da camada física. O formato para este VPI adicional ainda não foi determinado.
- **Identificador de circuito virtual (Virtual Path Identifier – VPI).** O VPI é um campo de 8-bits numa célula UNI e um campo de 12-bits numa célula NNI (veja ilustração anterior).
- **Identificador do canal virtual (Virtual Channel Identifier – VCI).** O campo VCI possui 16-bits tanto nos frames UNI quanto nos NNI.
- **Tipo de payload (Payload Type – PT).** O campo PT possui três bits. O primeiro bit define o payload como dados do usuário ou informação de gerenciamento. A interpretação dos dois últimos bits depende do valor do primeiro bit.
- **Controle de congestionamento (Cell Loss Priority – CLP).** O campo CLP de 1-bit destina-se ao controle de congestionamento. Este campo indica a prioridade para o descarte de células pelos comutadores. O valor CLP=1 para uma célula implica em que, caso o nó tenha que descartar, esta célula será descartada primeiro.
- **Correção de erro do cabeçalho (Header Error Correction – HEC).** O HEC é um código calculado a partir dos quatro primeiros bytes do cabeçalho. Este é um CRC com divisor $x^8 + x^2 + x + 1$ que é utilizado para corrigir erros simples e uma enorme classe de erros múltiplos.

Camada de Adaptação (Application Adaptation Layer – AAL)

A camada de adaptação ATM (AAL) foi desenvolvida para habilitar dois conceitos ATM. Primeiro, o padrão ATM deve aceitar qualquer tipo de payload, tanto frames de dados quanto cadeias de bits. Um frame de dados pode chegar do protocolo de camada superior que montou o frame a ser enviado através de uma rede tal como ATM. Um bom exemplo é a Internet. O protocolo ATM também é capaz de transportar payload de multimídia. Ele pode receber uma cadeia de bits e segmentá-los em tamanhos menores capazes de serem encapsulados em células na camada ATM. A camada AAL utiliza duas subcamadas para tal finalidade.

O payload deve ser segmentado em grupos de 48-bytes para ser transportado, independentemente se os dados estiverem num frame de dados ou numa cadeia de bits. No destino, tais segmentos necessitam de reagrupamento para recriar o payload original. A camada AAL define uma subcamada, denominada **subcamada de segmentação e reagrupamento (Segmentation and Reassembly – SAR)**. A segmentação acontece na origem e o reagrupamento é feito no destino.

Depois da segmentação realizada pela subcamada SAR, os dados devem ser preparados de modo a assegurar a integridade da transmissão. Isto é realizado na **subcamada de convergência de transmissão (Convergence Sublayer – CS)**.

O protocolo ATM define quatro versões para a AAL: **AAL1, AAL2, AAL3/4 e AAL5**.

AAL1 Uma das quatro AALs recomendadas pelo ITU-T. A AAL1 é usada para serviços orientados à conexão sensíveis ao atraso que exigem taxas de bits constantes, como, por exemplo, vídeo descompactado e outros tráfegos isócronos. Ela possibilita que o protocolo ATM utilize, por exemplo, os recursos do canal de voz (rede de telefonia pública fixa) ou de linhas T. A Figura 18.27 ilustra como uma cadeia de bits de dados é segmentada em grupos de 47-bytes e encapsulados nas células.

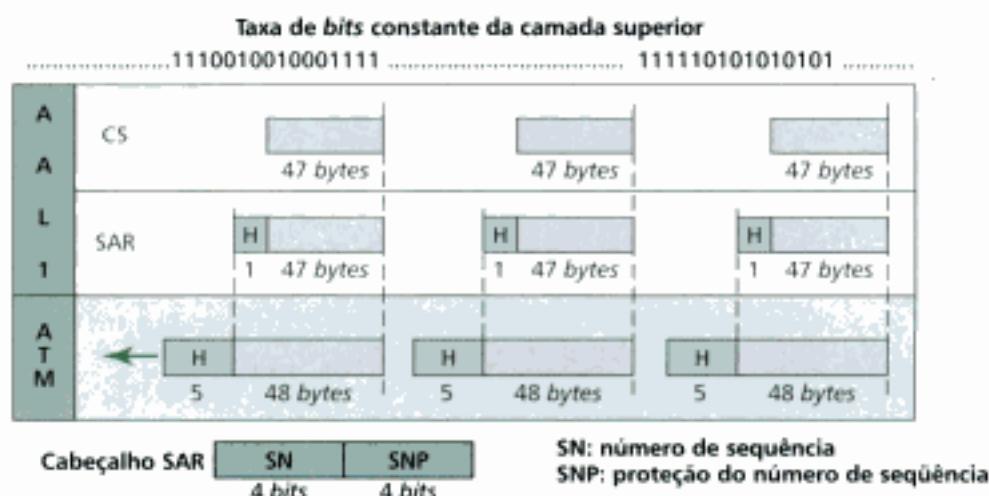


Figura 18.27 AAL1.

A subcamada CS divide a cadeia de *bits* em segmentos de 47-bytes e os repassa à subcamada SAR localizada imediatamente abaixo. Perceba que a subcamada CS não adiciona cabeçalho.

A subcamada SAR adiciona 1-byte ao cabeçalho e repassa o segmento de 48-bytes à camada ATM. O cabeçalho possui dois campos:

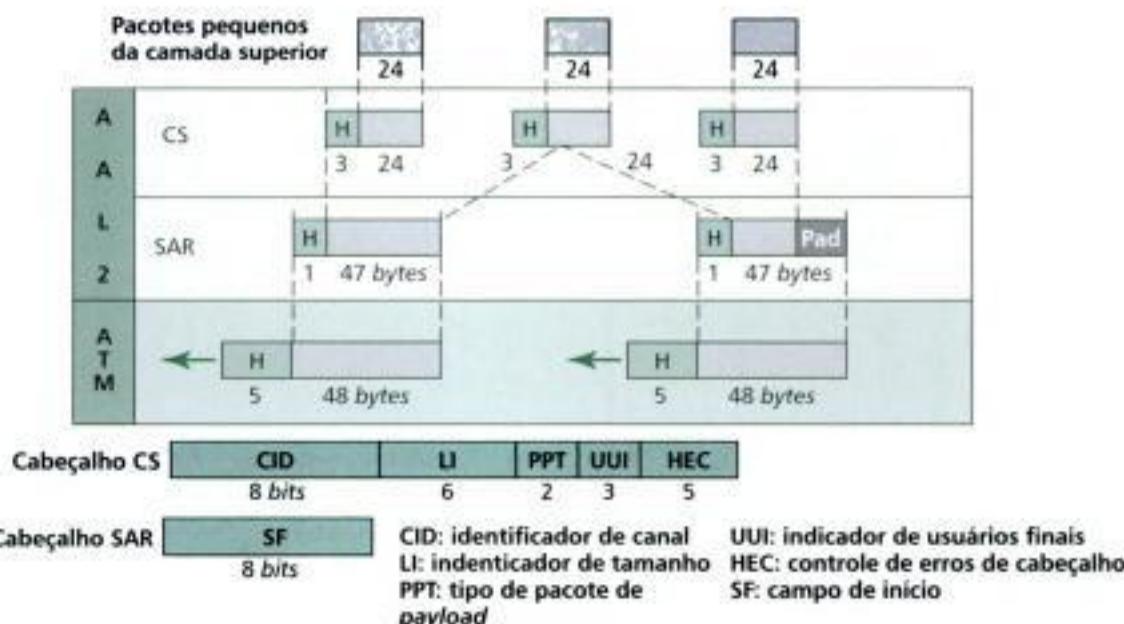
- **Número de seqüência (Sequence Number – SN).** Este campo de 4-bits define um número de seqüência para ordenar os *bits*. Às vezes, o primeiro bit é utilizado para temporização e os outros três bits são deixados para o número seqüencial (módulo 8).
- **Proteção do número de seqüência (Sequence Number Protection – SNP).** O segundo campo de 4-bits adiciona a correção do primeiro campo. Os três primeiros bits corrigem automaticamente o campo SN. O último é um bit de paridade que detecta erro em todos os 8 bits.

AAL2 A subcamada AAL2 foi desenvolvida originalmente para suportar uma cadeia de *bits* de dados ininterrupta. Entretanto, a AAL2 foi reprojeta de modo a receber novas atribuições. Hoje, a AAL2 é usada para serviços orientados à conexão que suportam uma taxa de *bits* variável, como, por exemplo, alguns tipos de tráfego isócrono de vídeo e voz (com e sem compressão). Um exemplo interessante de utilização da subcamada AAL2 acontece na telefonia móvel. A subcamada AAL2 permite a multiplexação de pequenos *frames* para dentro das células.

A Figura 18.28 apresenta o processo de encapsulamento de pequenos *frames* de uma origem única (o usuário de telefonia móvel) ou de muitas origens (muitos usuários de telefonia móvel) dentro de uma célula.

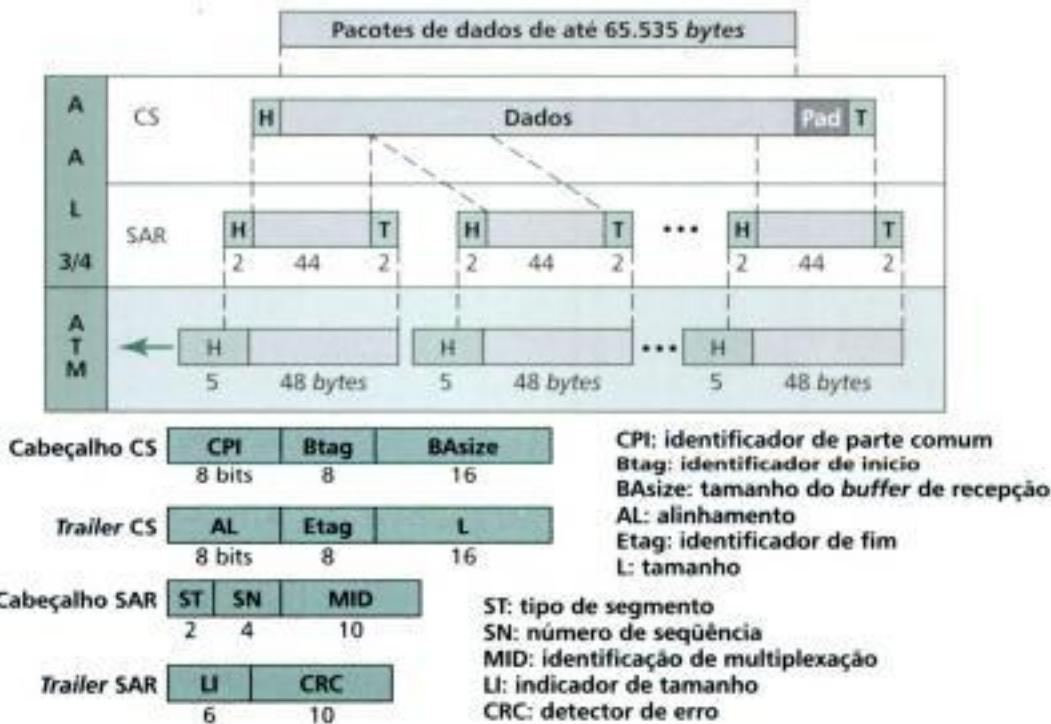
O *overhead* da subcamada CS é constituído de cinco campos:

- **Identificador de canal (Channel Identifier – CID).** O CID possui 8-bits e define o canal (usuário) do pacote de dados pequeno.
- **Indicador de tamanho (Length Indicator – LI).** O campo LI possui 6-bits e indica quanto do pacote final representa dados.
- **Tipo de pacote de payload (Packet Payload Type – PPT).** O campo PPT define o tipo de pacote a ser encapsulado na subcamada AAL2.
- **Indicador de usuários finais (User-to-User Indicator – UUI).** O campo UUI pode ser utilizado por usuários finais.
- **Controle de erros do cabeçalho (Header Error Control – HEC).** Os últimos 5-bits são utilizados para correção de erros no cabeçalho.

**Figura 18.28** AAL2.

O único tipo de *overhead* gerado na camada SAR é o campo de início (Start Field – SF) que define o deslocamento do início do pacote.

AAL3/4 Inicialmente, a AAL3 foi planejada para suportar serviços de dados orientados à conexão e a AAL4 para suportar serviços sem conexão. Entretanto, ocorreu um agrupamento dessas duas camadas de adaptação inicialmente distintas (recomendadas pelo ITU-T). Hoje a AAL3/4 suporta tanto os *links* sem conexões como os orientados à conexão, mas é usada principalmente para a transmissão de pacotes SMDS (Switched Multimegabit Data Service) através das redes ATM.

**Figura 18.29** AAL3/4.

O cabeçalho (*header*) e o rótulo (*trailer*) dos *frames* da subcamada CS são constituídos de seis campos:

- **Identificador de parte comum (Common Part Identifier – CPI).** O campo CPI define o modo como os campos subsequentes serão interpretados. O valor padrão é 0.
- **Identificador de início (Begin Tag – Btag).** O valor deste campo é repetido em cada célula de modo a identificar todas as células oriundas do mesmo pacote. O valor é o mesmo do campo Etag (veja descrição abaixo).
- **Tamanho do buffer de recepção (Buffer Allocation Size – BAsize).** O campo BAsize possui dois bytes de extensão para informar ao receptor que tamanho de buffer é necessário alocar para receber os dados de entrada.
- **Alinhamento (AL).** O campo de 1-byte AL é incluído para aumentar o tamanho do resto do trailer para 4-bytes.
- **Identificador de fim (Etag).** O campo ET de 1-byte funciona como uma flag de término. O valor desse campo é o mesmo do Btag.
- **Length (Comprimento).** Este campo de 2-bytes indica o comprimento da unidade de dados (*payload*).

O cabeçalho e o rótulo da subcamada SAR são divididos em cinco campos:

- **Tipo de segmento (Segment Type – ST).** Este identificador de 2-bits especifica a posição do segmento na mensagem: início (00), meio (01) e fim (10). Um único segmento de mensagem possui o ST igual a 11.
- **Número de seqüência (Sequence Number – SN).** Este campo já foi definido na subcamada AAL1.
- **Identificação de multiplexação (Multiplexing Identification – MID).** O campo MID de 10-bits identifica as células oriundas de fluxos de dados diferentes, multiplexadas na mesma conexão virtual.
- **Identificador de comprimento (Length Indicator – LI).** Este campo define o quanto do pacote representa dados (*payload*) e não bits de enchimento (*padding*).
- **CRC.** Os últimos 10-bits do rótulo é um campo de CRC para toda unidade de dados.

AAL5 A camada AAL5 suporta serviços orientados à conexão e é usada predominantemente para transferência do IP clássico por tráfego ATM e LANE (LAN Emulation). A AAL5 usa a **SEAL (Simple and Efficient Adaptation Layer)** e é a menos complexa das recomendações AAL atuais. Oferece sobrecarga de largura de banda baixa e requisitos de processamento mais simples em troca da capacidade de largura de banda reduzida e do recurso de recuperação de erro. A **AAL5** assume que todas as células pertencentes a uma mesma mensagem viajam seqüencialmente e que as funções de controle estão incluídas nas camadas superiores da aplicação destino. A Figura 18.30 mostra a subcamada AAL5.

Os quatro campos de rótulos (*trailers*) da camada CS são:

- **Usuários finais (User-to-User – UU).** Este campo é utilizado pelos usuários finais, como já foi descrito na subcamada AAL2.
- **Identificador de parte comum (CPI).** Este campo já foi definido anteriormente.
- **Tamanho (Length – L).** O campo L possui 2-bytes para indicar o comprimento original dos dados.
- **CRC.** Os últimos 4-bytes são para controle de correção de erros da unidade de dados.

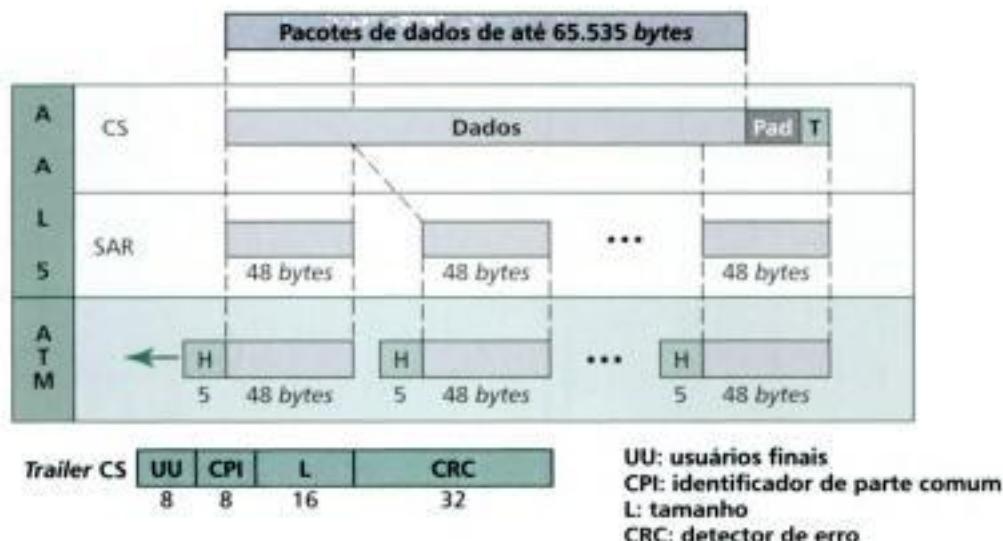


Figura 18.30 AAL5.

Controle de Congestionamento e Qualidade de Serviços

O protocolo ATM possui mecanismos muito desenvolvidos de controle de congestionamento e qualidade de serviços. Discutiremos tais mecanismos no Capítulo 23.

LANs ATM

Muitos esforços tem sido realizados para aplicar a tecnologia ATM a redes LANs. O resultado é a LAN ATM. No Apêndice G discutimos um pouco sobre as LANs ATM.

18.4 TERMOS-CHAVE

AAL1	Fase de estabelecimento
AAL2	Fase de transferência de dados
AAL3/4	Frame Relay
AAL5	Frame Relay Assembler/Disassembler (FRAD)
Asynchronous Transfer Mode (ATM)	Identificador de caminho virtual (VPI)
Banda sob demanda	Identificador de circuito virtual (VCI)
Camada ATM	Local Management Information (LMI)
Camada de adaptação ATM (AAL)	Network-to-Network Interface (NNI)
Caminho de transmissão (TP)	Qualidade de serviços (QoS)
Caminho virtual (VP)	Rajada de dados
Células	Rede de células
Círculo virtual (VC)	Segmentação e reagrupamento (SAR)
Círculo virtual comutado (SVC)	Simple and Efficient Adaptation Layer (SEAL)
Círculo virtual permanente (PVC)	Subcamada de Convergência (CS)
Comutação de células	User-to-Network Interface (UNI)
Comutação de circuitos virtuais	Voz sobre Frame Relay (Voice Over Frame Relay – VOFR)
Controle de congestionamento	X.25
Data Link Connection Identifier (DLCI)	
Fase de desconexão	

18.5 RESUMO

- Comutação de circuitos virtuais é uma tecnologia da camada de enlace para permitir o compartilhamento de *links*.
- Um identificador de circuito virtual (VCI) rotula um *frame* entre dois nós de comunicação.

- As três fases do processo de comunicação numa rede de comutação de circuitos virtuais são: estabelecimento, transferência de dados e desconexão.
- A fase de estabelecimento pode utilizar um circuito virtual permanente (PVC) ou um circuito virtual comutado (SVC).
- Frame Relay é uma tecnologia relativamente rápida e de baixo custo para comunicação em rajada de dados.
- Tanto as conexões PVC quanto as SVC são utilizadas no Frame Relay.
- O identificador DLCI (Data Link Connection Identifier) identifica um circuito virtual no protocolo Frame Relay.
- Asynchronous Transfer Mode (ATM) é um protocolo de comutação de células que em combinação com redes de alto desempenho, como a SONET, permite conexões de alta velocidade.
- Uma célula é um bloco de dados pequeno e de tamanho fixo.
- O pacote de dados ATM é uma célula composta de 53-bytes (5 bytes de cabeçalho e 48 bytes de payload).
- A tecnologia ATM elimina os tempos de atraso variáveis associados aos pacotes de tamanhos variáveis.
- A tecnologia ATM pode controlar transmissões em tempo real.
- A interface UNI (User-to-Network Interface) conecta um usuário a um nó de comutação ATM.
- A interface NNI (Network-to-Network Interface) conecta dois nós de comutação numa rede ATM.
- Numa rede ATM, a conexão entre dois dispositivos finais é realizada através dos caminhos de transmissão (TPs), caminhos virtuais (VPs) e circuitos virtuais (VCs).
- Nas redes ATM, uma combinação de identificador de caminho virtual (VPI) e identificador de circuito virtual (VCI) mapeia uma conexão virtual.
- O padrão ATM define três camadas:
 - Camada de adaptação ATM (AAL) recebe as transmissões dos serviços das camadas mais altas e os mapeia em células ATM.
 - A camada ATM provê serviços de roteamento, gerenciamento de tráfego, comutação e de multiplexação.
 - A camada física define o meio de transmissão, a transmissão de *bits*, codificação e conversão elétrica-óptica (redes SONET).
- A camada AAL é dividida em duas subcamadas: subcamada de convergência (CS) e subcamada de segmentação e reagrupamento (SAR).
- Existem quatro diferentes tipos de AALs, cada uma para um tipo específico de dados:
 - AAL1 para transmissões que exigem taxas de *bits* constantes.
 - AAL2 para pacotes pequenos.
 - AAL3/4 para comutação de pacotes convencionais (círculo virtual ou datagramas).
 - AAL5 para pacotes que não requerem seqüenciamento e mecanismos de controle de erros.

18.6 PRATIQUE OS CONHECIMENTOS ADQUIRIDOS

Questões de Revisão

- Compare o formato de um *frame* do protocolo HDLC com o *frame* do protocolo Frame Relay. Quais campos estão faltando no *frame* do protocolo Frame Relay? Quais campos foram adicionados ao protocolo Frame Relay?
- Por que o campo de controle do protocolo HDLC foi retirado do protocolo Frame Relay?
- O protocolo HDLC utiliza três tipos de *frames* (*I-frame*, *S-frame* e *U-frame*).

Qual deles corresponde ao *frame* do protocolo Frame Relay?

- Não há números de seqüência no protocolo Frame Relay. Por quê?
- Podemos conectar dois dispositivos a uma mesma rede Frame Relay usando os mesmos DLCIs?
- Por que o Frame Relay é uma solução alternativa melhor para conectar LANs que as linhas T?
- Compare um SVC com um PVC.

8. Examine a camada física do protocolo Frame Relay.
9. Por que a multiplexação torna-se mais eficiente quando as unidades de dados têm o mesmo tamanho?
10. Qual a diferença entre NNI e UNI?
11. Qual é o relacionamento entre TPs, VPs e VCs?
12. De que forma uma conexão virtual é identificada numa rede ATM?
13. Cite os nomes e as respectivas funções das camadas ATM.

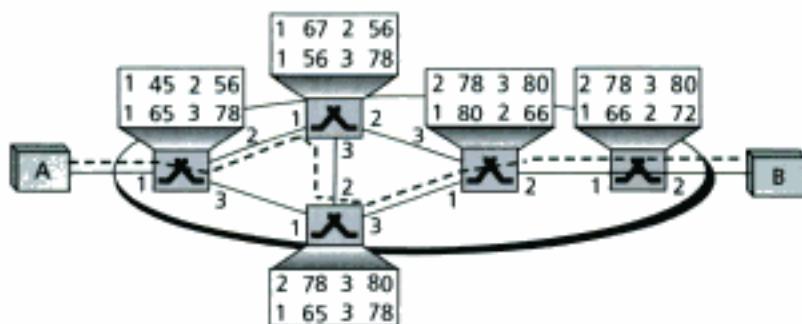
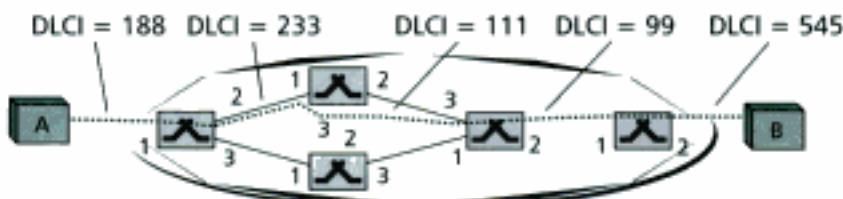
Questões de Múltipla Escolha

14. Frame Relay opera na camada _____.
 a. Física
 b. De enlace
 c. Física e de enlace
 d. Física, de enlace e de rede.
15. Na camada de enlace o Frame Relay utiliza _____.
 a. Protocolo BSC
 b. Um protocolo HDLC simplificado
 c. LAPB
 d. Qualquer protocolo padrão ANSI.
16. As funções de roteamento e de comutação das redes Frame Relay são realizadas pela camada _____.
 a. Física
 b. De enlace
 c. Rede
 d. (b) e (c)
17. As redes Frame Relay são inadequadas para _____ devido aos possíveis atrasos nas transmissões provocados por *frames* de tamanhos variáveis.
 a. Vídeo em tempo real
 b. Transferências de arquivo
 c. Comunicação a taxa de *bits* constantes
 d. Todas as respostas anteriores
18. O protocolo Frame Relay provê conexões _____.
 a. PVC
 b. SVC
 c. (a) e (b)
 d. Nenhuma das alternativas anteriores
19. O campo de endereço Frame Relay tem _____ de tamanho.
 a. 4-bytes
 b. 2-bytes
 c. 3-bytes
 d. Nenhuma das alternativas anteriores
20. Um dispositivo denominado _____ permite que *frames* de uma rede ATM sejam transmitidos através de uma rede Frame Relay.
 a. LMI
 b. VOFR
 c. FRAD
 d. DLCI
21. _____ é um protocolo de controle e interface de gerenciamento nas redes Frame Relay.
 a. LMI
 b. VOFR
 c. FRAD
 d. DLCI
22. _____ é uma opção Frame Relay que transmite voz através da rede.
 a. LMI
 b. VOFR
 c. FRAD
 d. DLCI
23. Na comunicação de dados, ATM é o acrônimo para _____.
 a. Automated Teller Machine
 b. Automatic Transmission Model
 c. Asynchronous Telecommunication Method
 d. Asynchronous Transfer Mode
24. Visto que ATM _____, o que significa que as células seguem o mesmo caminho, as células não chegam usualmente fora de ordem.
 a. É assíncrona
 b. É multiplexada
 c. É uma rede
 d. Utiliza roteamento de circuitos virtuais
25. Que camada do protocolo ATM reformata os dados recebidos de outras redes?
 a. Física
 b. ATM
 c. Adaptação
 d. Adaptação de dados

26. Que camada no protocolo ATM produz uma célula com 53-bytes como produto final?
- Física
 - ATM
 - Adaptação
 - Conversão de células
27. Que tipo de subcamada AAL foi projetada para suportar uma cadeia de dados a uma taxa de *bits* constante?
- AAL1
 - AAL2
 - AAL3/4
 - AAL5
28. Que tipo de subcamada AAL foi projetada para suportar SEAL?
- AAL1
 - AAL2
 - AAL3/4
 - AAL5
29. Numa rede ATM, todas as células pertencentes a uma mesma mensagem seguem o mesmo _____ e permanece na ordem original até que elas cheguem ao destino final.
- Caminho de transmissão
 - Caminho virtual
 - Círculo virtual
 - Nenhuma das alternativas anteriores
30. Um _____ provê uma conexão ou um conjunto de conexões entre nós de comutação.
- Caminho de transmissão
 - Caminho virtual
 - Círculo virtual
 - Nenhuma das alternativas anteriores
31. Um _____ é a conexão física entre um dispositivo final e um nó de comutação ou entre dois nós de comutação.
- Caminho de transmissão
 - Caminho virtual
 - Círculo virtual
 - Nenhuma das alternativas anteriores
32. O VPI de uma UNI tem _____ bits de tamanho.
- 8
 - 12
 - 16
 - 24
33. O VPI de um NNI tem _____ bits de tamanho.
- 8
 - 12
 - 16
 - 24

Exercícios

34. O campo de endereço de um *frame* do protocolo Frame Relay é 1011000100010110. Qual é o identificador DLCI (em decimal)?
35. O campo de endereço de um *frame* do protocolo Frame Relay é 101100000101001. Este endereço é válido?
36. Determine o valor DLCI se os três primeiros *bytes* recebidos são 7C 74 E1 (em hexadecimal).
37. Determine o valor dos 2-*bytes* do campo de endereço em hexadecimal se o valor DLCI é 178. Assuma que não há congestionamento de tráfego.
38. Na Figura 18.31, uma conexão virtual é estabelecida entre A e B. Mostre o valor DLCI para cada *link*.
39. Na Figura 18.32, uma conexão virtual é estabelecida entre A e B. Mostre as entradas correspondentes nas tabelas de comutação de cada nó.
40. Uma camada AAL1 recebe dados a 2 Mbps. Quantas células são criadas por segundo na camada ATM?
41. Qual é a eficiência total de uma rede ATM usando a camada AAL1 (a razão entre *bits* recebidos e *bits* transmitidos)?
42. Se uma aplicação utiliza a camada AAL3/4 e estão chegando 47.787 *bytes* de dados da camada CS, quantos *bytes* de *padding* são necessários? Quantas unidades de dados passaram da camada SAR para a camada ATM? Quantas células foram produzidas?
43. A eficiência de uma rede ATM usando AAL3/4 depende do tamanho do pacote? Explique sua resposta.
44. Qual é o número mínimo de células resultantes da entrada de um pacote na camada AAL3/4? Qual é o número máximo de células resultantes de um pacote de entrada?

**Figura 18.31** Exercício 38.**Figura 18.32** Exercício 39.

45. Qual é o número mínimo de células resultantes de entrada em um pacote da camada AAL5? Qual é o número máximo de células resultantes de um pacote de entrada?
46. Explique por que o processo de enchimento (*padding*) é desnecessário na camada AAL1, mas é necessário nas outras AALs?
47. Usando AAL3/4, mostre a situação onde precisamos _____ de *padding*.
 - 0 bytes (sem *padding*)
 - 40 bytes
 - 43 bytes
48. Usando AAL5, mostre a situação onde precisamos _____ de *padding*.
 - 0 bytes (sem *padding*)
- b. 40 bytes
- c. 47 bytes
49. Numa célula de 53-bytes, quantos bytes pertencem ao usuário nas seguintes camadas (sem *padding*)?
 - AAL1
 - AAL2
 - AAL3/4 (sem a primeira ou última célula)
 - AAL5 (sem a primeira ou última célula)
50. Complete a Tabela 18.1 entrando com o tamanho das unidades de dados na subcamada SAR para todas as AALs.
51. Quantas conexões virtuais podem ser definidas numa UNI? E numa NNI?

TABELA 18.1 Exercício 50

Subcamada	<i>AAL1</i>	<i>AAL2</i>	<i>AAL3/4</i>	<i>AAL5</i>
SAR				

PARTES

CAMADA DE REDE

Na arquitetura TCP/IP (modelo Internet) a camada de rede fornece conectividade e seleção de caminho entre dois sistemas finais. A camada de rede é a camada onde ocorre o roteamento. Ela é responsável pelos processos *host-to-host*. Noutras palavras, quando desejamos transmitir um pacote de San Francisco para Miami, há cooperação entre os protocolos das duas camadas de rede dos dois computadores para supervisionar a entrega da mensagem.

A Figura 1 mostra a posição da camada de rede no modelo de cinco camadas da Internet. A camada de rede é a terceira do modelo. Ela recebe serviços da camada de enlace e oferece serviços à camada de transporte. O principal serviço da camada de enlace é a entrega de dados entre nós da rede. Se existirem N nós entre os *hosts* de origem e destino, haverá N processos entre os nós, a serem realizadas *host-to-host* na camada de rede.

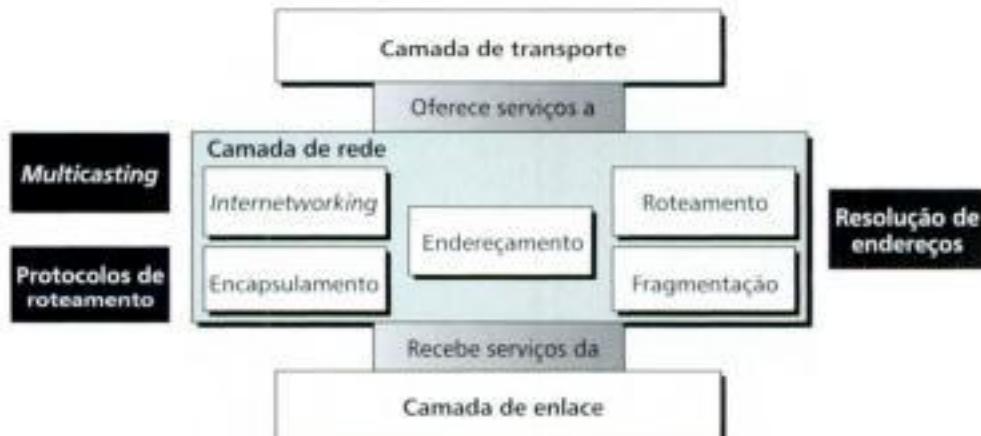


Figura 1 Posição da camada de rede.

Dois *hosts* estão freqüentemente separados através de muitas redes físicas. A camada de enlace é responsável pelo transporte de dados através da rede física; já a camada de rede ou camada de *internetwork*, como às vezes é denominada, é responsável pelo roteamento de pacotes através da Internet, possivelmente, através de uma infinidade de redes físicas.

Serviços

Conforme ilustra a Figura 2, a camada de rede oferece uma série de serviços.



Figura 2 Serviços da camada de rede.

Internetworking

O principal serviço da camada de rede é prover *internetworking*, a interconexão lógica de diversas redes físicas heterogêneas de modo que, para as camadas superiores (transporte e aplicação), essas redes comportem-se como uma única rede.

Endereçamento

Na camada de rede identificamos univocamente cada dispositivo da Internet de modo a permitir comunicação global entre todos os *hosts* da rede. Este esquema de endereçamento é análogo ao sistema telefônico, onde cada assinante possui um número de telefone (incluindo o código do país e o código de área). Por exemplo, o número de telefone 011 86 731 220 8098 identifica univocamente um número de telefone na cidade Changsha na província chinesa de Hunan.

Os endereços utilizados na camada de rede devem definir *única e universalmente* a conexão de um *host* (computador) ou roteador na Internet. Os endereços da camada de rede devem ser únicos para que cada um deles defina uma, e somente uma, conexão na Internet. Dois dispositivos na Internet nunca podem ter os mesmos endereços. Assim, se um dispositivo tiver duas conexões com a Internet (por exemplo, dois adaptadores de rede) ele terá dois endereços. Dedicaremos a maior parte do Capítulo 19 aos esquemas de endereçamento através da Internet.

Roteamento

Você já experimentou este dilema? Certamente isto já aconteceu com você quando quis sair de um determinado lugar e chegar noutro e, antes de conseguir, experimentou a desagradável situação de ser obrigado a escolher uma rota (entre muitas). Algumas opções seriam: uma certa rota é a mais curta, porém é a que está em piores condições. Outra é a mais longa, entretanto é a mais segura. Uma rota simplesmente o levará a um congestionamento durante horas ou então a um caminho onde terá de pagar pedágio. Outras seguem através de estradas em montanhas geladas ou perigosas.

Sempre que houver muitas rotas que levam a um mesmo destino devemos tomar uma decisão e escolher uma delas. Normalmente, nossa decisão baseia-se em algum critério que é, momentaneamente, importante para nós. Se estivermos dentro de um carro não muito confiável podemos escolher a estrada mais longa para evitar os riscos associados a uma estrada perigosa. Muitas vezes queremos escolher a estrada que minimiza o tempo de chegada. A Internet também é assim, uma combinação de inúmeras estradas através das quais o pacote IP viaja em direção a um destino particular e deve passar por diversas rotas. A diferença é que um pacote IP não pode escolher uma rota; os roteadores conectados às LANs e WANs tomam esta decisão. Examinaremos o roteamento no Capítulo 19.

Encapsulamento de Datagramas

A camada de rede encapsula os pacotes recebidos dos protocolos da camada superior (camada de transporte) em novos pacotes. No modelo da Internet, o encapsulamento é denominado datagrama ou pacote IP (Internet Protocol). No Capítulo 20, discutiremos este protocolo juntamente com outros protocolos da camada de rede.

Fragmentação

Um datagrama pode viajar por diferentes tipos de redes. Cada roteador desencapsula o datagrama IP do *frame* recebido, processa-o e, então, encapsula noutro *frame*. O formato e o tamanho do *frame* recebido depende do protocolo utilizado pela rede física da qual o *frame* originou. O formato e o tamanho de partida do *frame* depende do protocolo usado pela rede física para onde o *frame* está indo. Analisaremos a fragmentação no Capítulo 20 quando introduzirmos o protocolo IP.

Outros Aspectos

Existem outras questões não diretamente ligadas aos serviços da camada de rede, mas necessárias nesta parte do livro.

Resolução de Endereços

Quando um pacote precisa ser entregue num destino particular, ele precisa passar de um nó da rede para o próximo. A camada de rede oferece somente o endereçamento *host-to-host* e a camada de enlace precisa dos endereços físicos (MAC) para promover um processo entre os nós da rede. Existe um método de mapear estes dois endereços. Um protocolo denominado Address Resolution Protocol (ARP) pode fazer isto. Discutiremos esta aplicação no Capítulo 20.

Multicasting

Outra aplicação importante da Internet hoje é o *multicasting* (multidifusão), isto é, processos de transmissão de dados de um *host* para muitos. *Multicasting* tornou-se uma questão muito importante na Internet devido à crescente necessidade de recursos de multimídia na rede. Multimídia, na forma de áudio e vídeo, precisa de rotas *multicasting* para alcançar muitos destinos pertencentes, muitas vezes, a um grupo. Discutiremos as questões relativas ao *multicasting* no Capítulo 21.

Protocolos de Roteamento

Os protocolos de roteamento foram criados em resposta à demanda por tabelas de roteamento dinâmicas. Um protocolo de roteamento é uma combinação de regras e procedimentos escritos de modo a permitir que os roteadores troquem, uns com os outros, as tabelas de roteamento individuais. Os protocolos de roteamento permitem que os roteadores compartilhem todo o conhecimento adquirido sobre a internet e/ou a vizinhança deles. Através do compartilhamento de informação um roteador em San Francisco pode tomar conhecimento da falha de uma rede no Texas e escolher um outro caminho, caso haja, para direcionar os pacotes destinados à região de destino. Os protocolos de roteamento também incluem procedimentos para combinar informações recebidas de outros roteadores. Discutiremos os protocolos de roteamento no Capítulo 21.

Outros Protocolos de Suporte

Um protocolo de *internetworking* como o IP precisa do suporte de outro protocolo para ajudá-lo nos processos *host-to-host*. Este protocolo, denominado Internet Control Message Protocol (ICMP), será analisado no Capítulo 20.

Organização dos Capítulos

Incluímos três Capítulos nesta parte do livro. O Capítulo 19 discute o conceito geral a respeito da camada de rede: comunicação processo a processo e roteamento. O Capítulo 20 cobre os protocolos de *internetworking* da Internet: ARP, IP, ICMP e IGMP. O Capítulo 21 é dedicado aos protocolos de roteamento *unicast* e *multicast*.

Processos Host-to-Host: Internetworking, Endereçamento e Roteamento

Neste capítulo discutiremos a questão essencial relacionada à camada de rede. Precisaremos compreender inicialmente o conceito de *internetworking* e a importância da camada de rede para o processo *host-to-host*. Analisaremos também alguns tipos de comutação utilizados na Internet. Mostraremos que a Internet, no nível da camada de rede, é uma rede de comutação de pacotes não orientada à conexão.

Qualquer *internetwork*, em particular a Internet, necessita de um esquema de endereçamento global. Examinaremos tais esquemas utilizados na Internet e as questões relacionadas ao endereçamento.

Todos concordam que o roteamento é a questão mais importante e complexa a ser tratada na esfera da Internet. O roteamento é necessário para garantir que um pacote de dados oriundo do canto mais afastado no mundo, conectado à Internet, alcance um escritório localizado no Silicon Valley (Vale do Silício) na Califórnia.

19.1 INTERNETWORKS

As camadas física e de enlace de dados de uma rede operam apenas localmente. Estas duas camadas são responsáveis pelos processos de transmissão de dados de um nó da rede para o próximo. A Figura 19.1 ilustra um exemplo de uma *internetworking*.

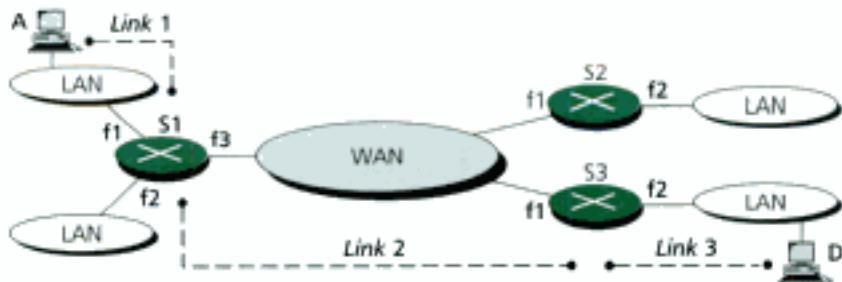


Figura 19.1 Internetwork.

A *internetwork* da figura é constituída de cinco redes: quatro LANs e uma WAN. Se o *host* A possui dados e quer transmiti-los ao *host* D, primeiro o *host* A deve encapsular os dados em

pacotes e enviá-los a S1 (um *switch* ou roteador). Por sua vez, S1 encaminha os pacotes a S3 e, finalmente, S3 processa a entrega ao *host* D. Dizemos que um pacote de dados passou através de três *links*.

Em cada *link* duas camadas físicas e duas camadas de enlace estão envolvidas, conforme ilustra a Figura 19.2.

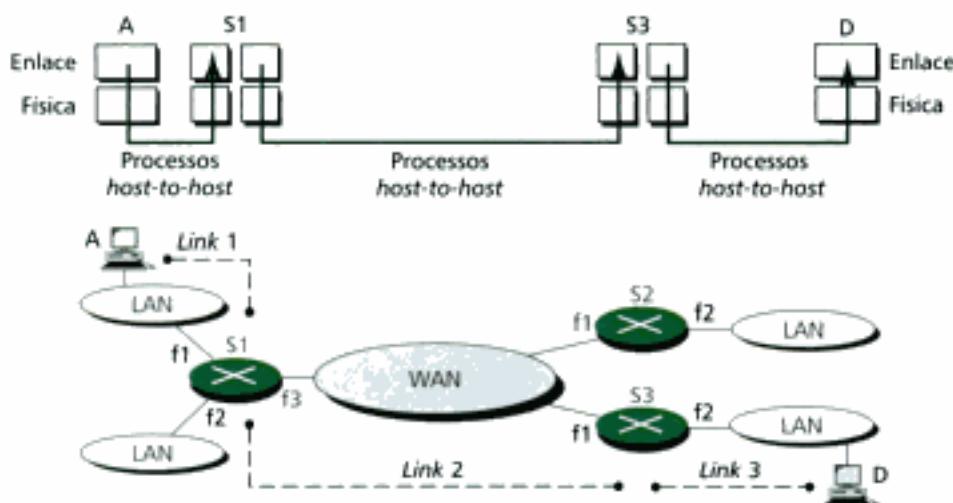


Figura 19.2 Links numa internetwork.

Entretanto, há um grande problema aqui. Quando os pacotes dados chegam na interface f1 do dispositivo S1, como S1 sabe que deve transmiti-los à interface f3 de S3? Não há nenhuma provisão na camada de enlace ou física que ajude S1 a tomar a decisão correta. Os pacotes também não carregam nenhuma informação sobre roteamento. Os pacotes contém os endereços MAC de origem A e MAC de destino S1. Para uma LAN ou uma WAN, um processo de envio de pacotes significa transportar um *frame* através de um *link*, não além dele.

Necessidade da Camada de Rede

A camada de rede ou camada de *internetworking*, como às vezes é chamada, foi projetada para resolver este problema de entrega através de múltiplos *links*. A camada de rede é responsável pelos processos *host-to-host* e pelo roteamento de pacotes através dos roteadores e/ou *switches*. A Figura 19.3 mostra a mesma *internetwork* com a camada de rede adicionada.

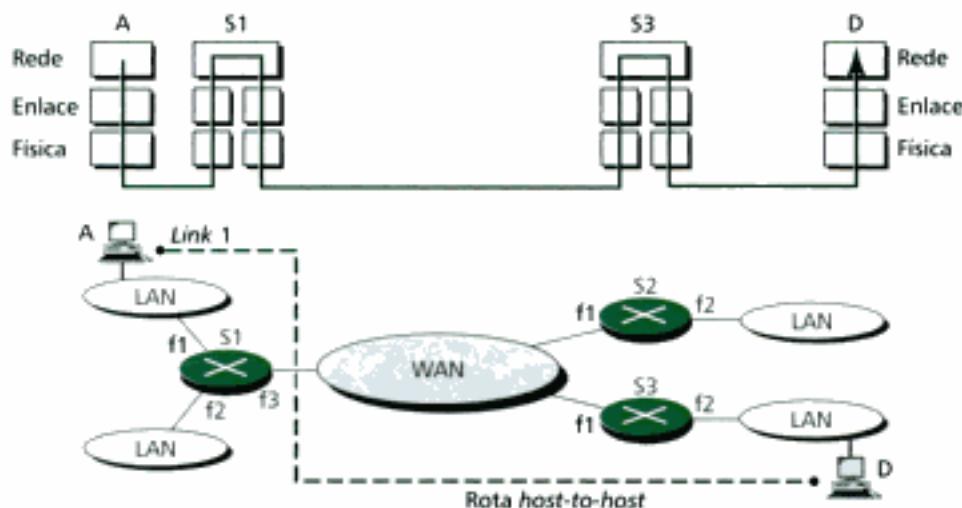


Figura 19.3 Camada de rede numa internetwork.

Hidden page

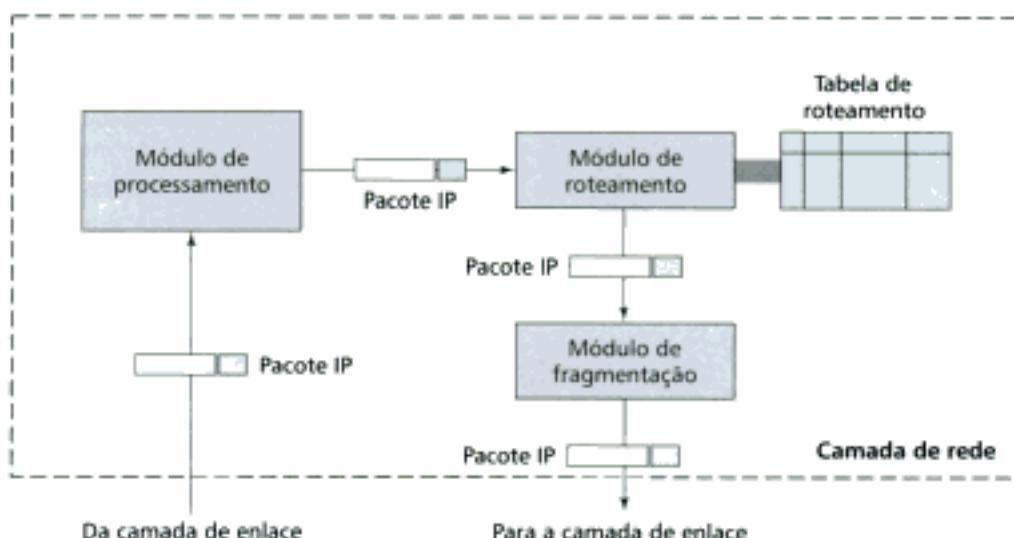


Figura 19.5 Camada de rede do roteador.

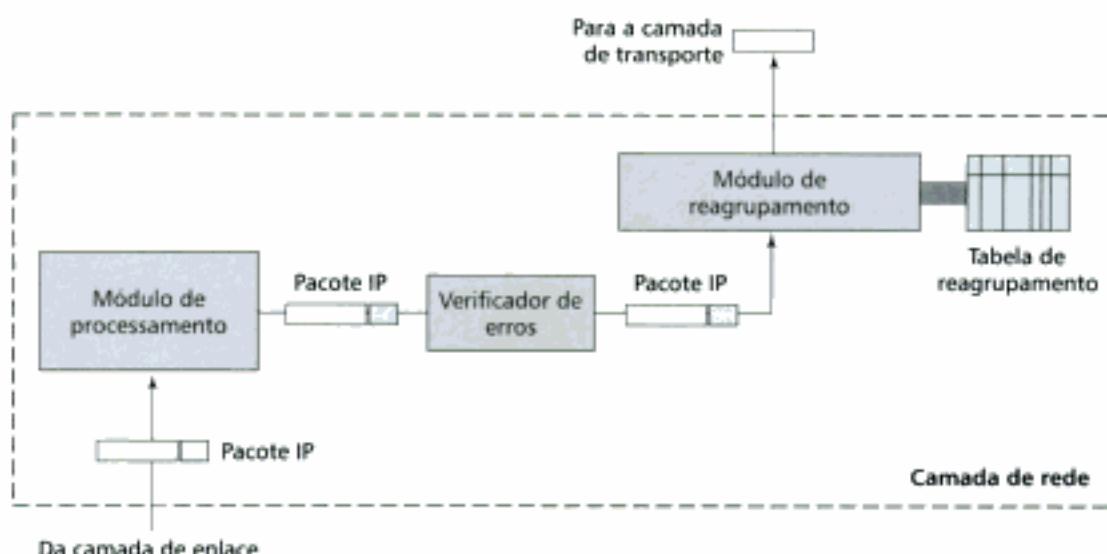


Figura 19.6 Camada de rede do *host* de destino.

categorias: comutação de circuitos e comutação de pacotes. A comutação de pacotes utiliza os mecanismos de circuitos virtuais ou de datagramas. A Figura 19.7 mostra a sistemática.

Na comutação de circuitos, um *link* físico é dedicado entre o *host* de origem e de destino. Neste caso, os dados podem ser enviados como uma cadeia de *bits* sem a necessidade de criação dos pacotes. Na **comutação de pacotes**, por outro lado, os dados são transmitidos em unidades discretas denominadas **pacotes** (potencialmente de tamanhos variáveis). Cada pacote contém não só dados mas também um cabeçalho com informação de controle (tal como os endereços do *host* de origem e do *host* de destino). Os pacotes são transmitidos pela rede, deslocando-se de nó em nó. Em cada nó, o pacote é armazenado brevemente antes de ser roteado de acordo com a informação contida no cabeçalho. Há duas técnicas comuns de comutação de pacotes: através de datagramas e de circuitos virtuais.

Circuitos Virtuais

Na **abordagem de circuito virtual** para comutação de pacotes, o relacionamento entre todos os pacotes pertencentes a uma mensagem ou sessão é preservado. Uma única rota é escolhida entre o transmissor e o receptor no início da sessão. Quando os dados são enviados, todos os pacotes da transmissão viajam um após o outro, através da uma única rota. As redes WANs utilizam circuitos

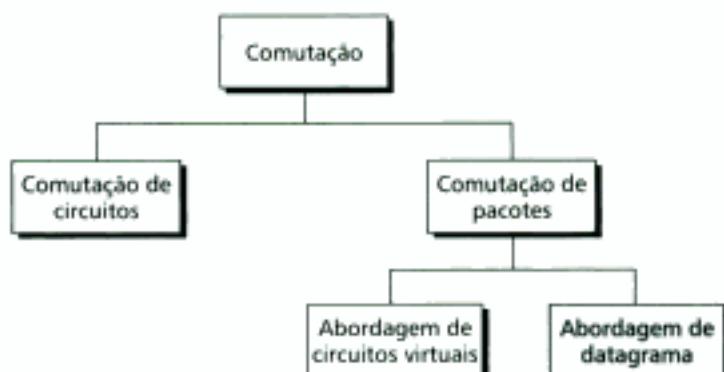


Figura 19.7 Comutação.

virtuais para implementar a comutação de pacotes. De acordo com o Capítulo 18, os circuitos virtuais necessitam da fase de estabelecimento do circuito virtual entre o *host* de origem e de destino. Após a fase de estabelecimento de conexão, o roteamento toma lugar baseado no identificador de circuito virtual (VCI). A fase de desconexão desfaz o circuito virtual entre os *hosts*. Esta técnica é utilizada nas redes WANs, Frame Relay e ATM e é implementada na camada de enlace.

Datagramas

Na **abordagem de datagramas** para a comutação de pacotes, cada pacote é tratado independentemente dos demais. Isto acontece até mesmo se o pacote faz parte de uma transmissão fragmentada, a rede trata cada pacote como se ele tivesse existência isolada. Os pacotes nesse tipo de abordagem são denominados **datagramas**.

A Figura 19.8 ilustra como a abordagem de datagramas pode ser utilizada na entrega de quatro pacotes do *host* A ao *host* X. Nesse exemplo, todos os pacotes ou datagramas, partes de uma única mensagem, seguem por caminhos diferentes até o destino final (*host* X).

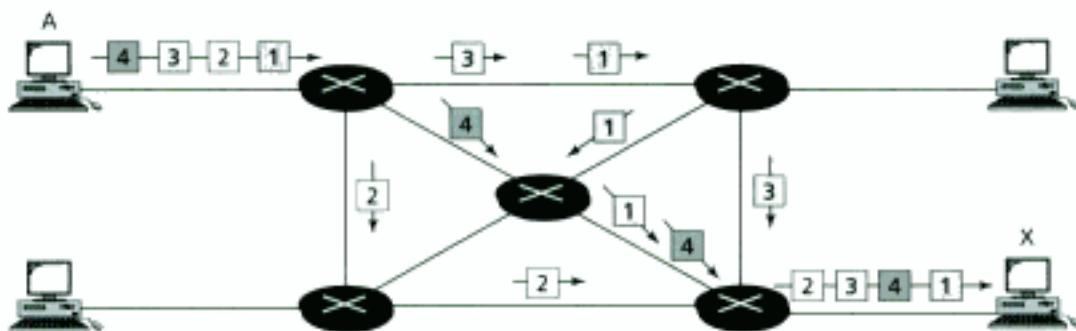


Figura 19.8 Abordagem de datagrama.

Esta abordagem geralmente provoca o recebimento de pacotes fora de ordem no *host* de destino. A maioria dos protocolos de rede deixa a responsabilidade de reordenação dos datagramas para a camada superior (camada de transporte) antes que os dados remontados sejam repassados para a aplicação que vai fazer uso deles.

A abordagem de datagramas também possui algumas vantagens. Ela não necessita da fase de estabelecimento de conexão e de identificadores de circuitos virtuais. O roteamento e o processo de entrega do pacote são baseados nos endereços de origem e de destino rotulados no cabeçalho do pacote. Os roteadores ou *switches* de camada 3 têm uma tabela de roteamento para serem utilizadas no roteamento dos dois endereços.

A Internet utiliza a abordagem de datagramas para comutação na camada de rede. Ela utiliza endereços universais definidos na camada de rede para rotear pacotes do *host* de origem ao *host* de destino.

Hidden page

Há duas notações correntes para o endereço IP: notação binária e a notação decimal com pontos.

Notação Binária

Na **notação binária**, o endereço IP é representado através de 32-bits. Para tornar este endereço um pouco mais inteligível, um ou mais espaços são inseridos entre cada octeto (8 bits). Nos referimos a cada octeto freqüentemente como um *byte*. Assim, é comum ouvirmos que o endereço IP é um endereço de 32-bits, um endereço de 4-octetos ou um endereço de 4-bytes. O endereço a seguir representa um endereço IP válido escrito na notação binária:

01110101 10010101 00011101 11101010

Notação Decimal com Pontos

Para tornar o endereço IP compacto e mais fácil de ser lido, os endereços de Internet são usualmente escritos com os *bytes* separados, escritos na forma decimal com pontos. A Figura 19.9 mostra um endereço IP escrito na **notação decimal com pontos**. Visto que cada *byte* (octeto) tem somente 8-bits, qualquer número na notação decimal com ponto deve ficar restrito à faixa entre 0 e 255.

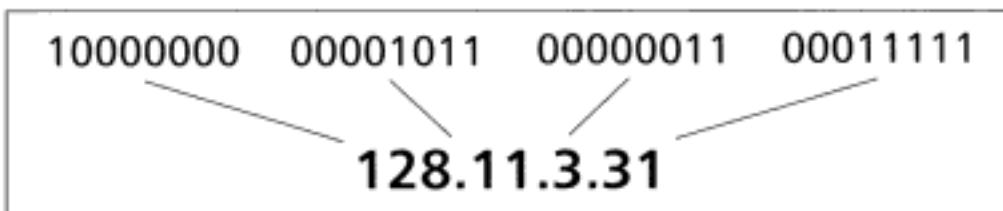


Figura 19.9 Notação decimal com pontos.

Os sistemas de numeração binário, decimal e hexadecimal são revisados no Apêndice B.

Exemplo 1

Escreva os seguintes endereços IP na notação decimal com pontos.

- 10000001 00001011 00001011 11101111
- 11111001 10011011 11111011 00001111

Solução

Para reescrevermos os endereços na notação decimal com pontos basta determinar o equivalente decimal de cada *byte* (veja o Apêndice B) e separá-los com pontos:

- 129.11.11.239
- 249.155.251.15

Exemplo 2

Escreva os seguintes endereços IP na notação binária.

- 111.56.45.78
- 75.45.34.78

Solução

Basta substituir cada número decimal pelo equivalente binário escrito em oito bits (veja Apêndice B).

- 01101111 00111000 00101101 01001110
- 01001011 00101101 00100010 01001110

Classes de Endereçamento

Os endereços IP, quando definidos algumas décadas atrás, usavam o conceito de classes. Esta arquitetura é denominada **classes de endereçamento**. Em meados da década de 90, uma nova arquitetura, denominada **endereçamento sem classes (Classless Addressing)** foi introduzida e, eventualmente, substituirá a arquitetura original. Entretanto, hoje a maioria dos endereços de Internet ainda utilizam as classes de endereçamento e a migração está acontecendo a passos lentos. Assim, precisamos discutir em profundidade tais classes.

Os endereços IP foram divididos em cinco classes de endereçamento: **classes A, B, C, D e E**. Cada classe ocupa uma parte ou uma **faixa de endereços IP**.

O endereço IP é dividido em cinco classes de endereçamento: A, B, C, D e E.

Tanto na notação binária quanto na decimal podemos determinar a classe a qual pertence um endereço IP.

Determinando a Classe na Notação Binária

Se o endereço IP aparecer escrito na notação binária, os *bits* mais significativos do primeiro *byte* revelam a classe de endereçamento a qual pertence o endereço em questão, como mostra a Figura 19.10.

	Primeiro byte	Segundo byte	Terceiro byte	Quarto byte
Classe A	0			
Classe B	10			
Classe C	110			
Classe D	1110			
Classe E	1111			

Figura 19.10 Determinando a classe de um endereço IP da notação binária.

Podemos seguir o procedimento sugerido na Figura 19.11 para verificar sistematicamente os bits e determinar a classe.

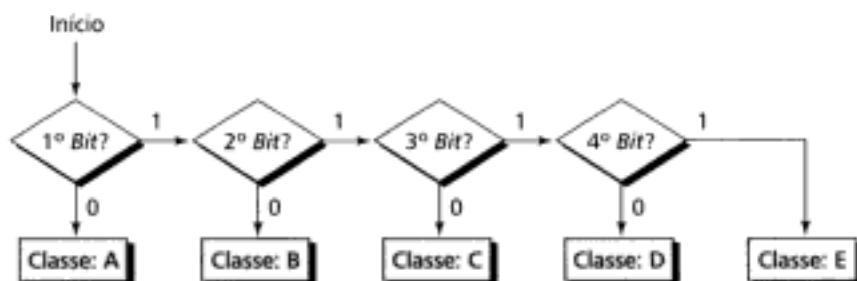


Figura 19.11 Determinando a classe de um endereço IP.

Exemplo 3

Determine a classe de cada um dos endereços abaixo:

- a. 00000001 00001011 00001011 11101111
- b. 11110011 10011011 11111011 00001111

Solução

Veja o procedimento da Figura 19.11.

Hidden page

	Byte 1	Byte 2	Byte 3	Byte 4
Classe A	<i>Netid</i>		<i>Hostid</i>	
Classe B		<i>Netid</i>		<i>Hostid</i>
Classe C		<i>Netid</i>		<i>Hostid</i>
Classe D			Endereço multicast	
Classe E			Reservado para uso futuro	

Figura 19.13 *Netid* e *hostid*.

Na classe A, um *byte* define a parte do endereço da rede (*netid*) e três *bytes* definem a parte de endereço do *host* (*hostid*). Na classe B, dois *bytes* definem tanto o *netid* quanto o *hostid*. Na classe C, três *bytes* definem o *netid* e um *byte* define o *hostid*.

Classes e Faixas

Um problema relacionado às classes de endereçamento é que cada classe é dividida em uma faixa de IPs fixos e, cada uma delas, têm um tamanho fixo. Vejamos cada classe cuidadosamente.

Classe A Esta classe é dividida em 128 endereços de redes diferentes. O primeiro endereço da faixa varia de 0.0.0.0 a 0.255.255.255 (*netid 0*). A segundo endereço da faixa varia de 1.0.0.0 a 1.255.255.255 (*netid 1*). O último endereço da faixa varia de 127.0.0.0 a 127.255.255.255 (*netid 127*). Perceba que cada um dos endereços especificados pelo primeiro *byte* (*netid*) são os mesmos, mas os outros 3-bytes (*hostid*) podem assumir qualquer valor na faixa. A Figura 19.14 mostra alguns endereços na classe A.

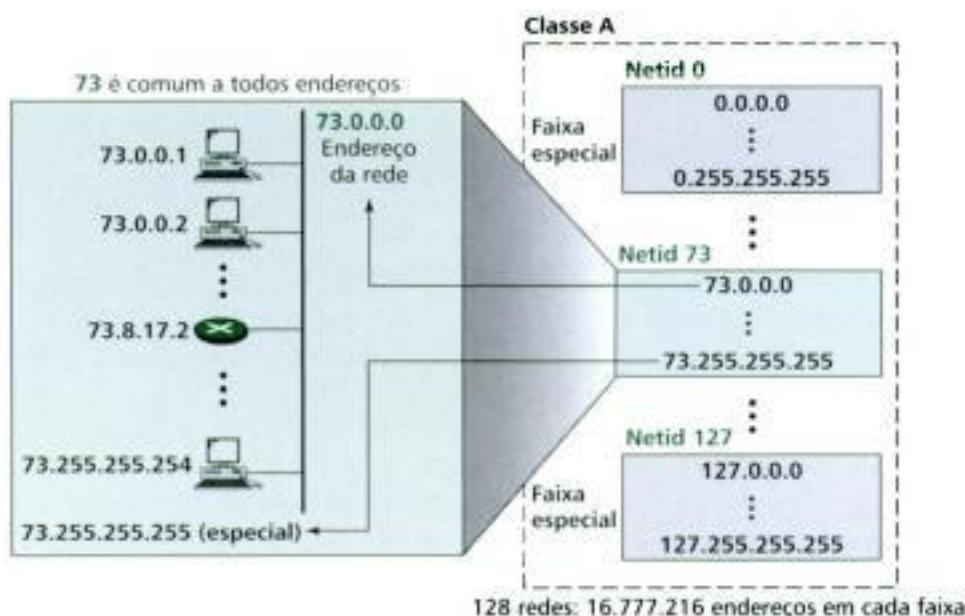


Figura 19.14 Faixa de endereços na classe A.

A Figura 19.14 também mostra como uma empresa, que recebeu o endereço de *netid* 73, usa os endereços concedidos a ela. O primeiro endereço da faixa é utilizado para identificar a empresa dentro da Internet. Este endereço é denominado **endereço da rede**. Ele define a rede da empresa, não um *host* particular dentro dela. A empresa não pode utilizar o último endereço. Ele é reservado para um propósito especial, como veremos em breve.

Os endereços da classe A foram criados para grandes empresas ou organizações com um número de *hosts* ou roteadores internos muito grande. Esta classe é muito utilizada por governos de

Hidden page

so, cada endereço nesta classe possui 256 endereços de *host*, o que significa que as empresas devem ser pequenas ou suficiente para possuírem menos que 256 endereços. A Figura 19.16 mostra a faixa de endereços IP na classe C.

Os endereços de classe C foram criados para serem utilizados em pequenas organizações ou empresas cuja quantidade de *hosts* na rede é pequeno. A quantidade de endereços de *hosts* na classe C é tão limitada que a maioria das empresas tem problemas com um endereço nesta classe.

A quantidade de endereços de *hosr* em uma rede classe C não atende às necessidades da maioria das empresas.

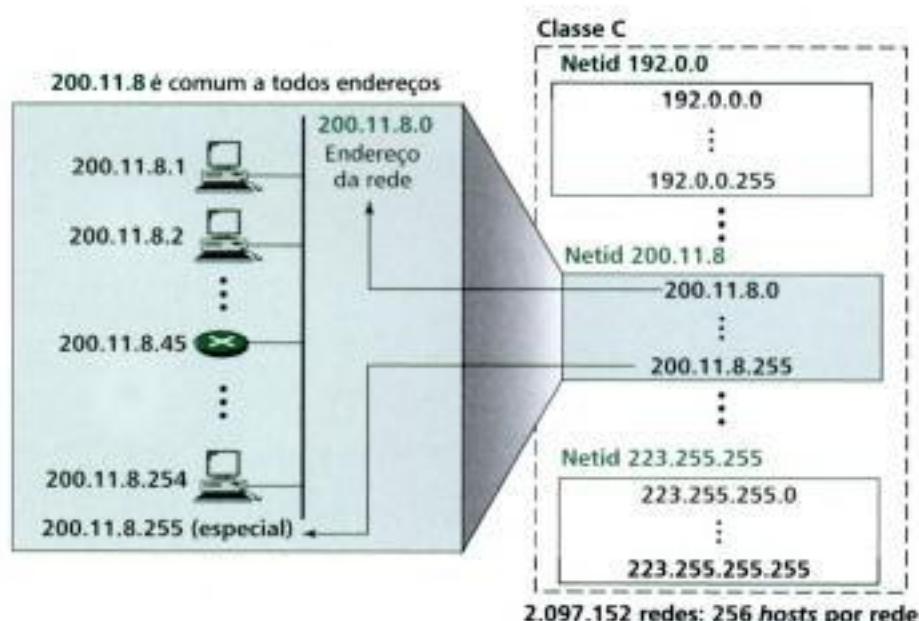


Figura 19.16 Faixas de endereços na classe C.

Classe D Há apenas uma faixa de endereços classe D criados para funções de *multicasting*. Estes endereços só podem ser utilizados para *hosts* de destino.

Classe E Há apenas uma faixa de endereços classe E. Ela foi criada para uso como endereços reservados.

Endereço da Rede

O endereço de rede é um endereço que define a rede de uma empresa ou organização dentro da Internet. Este endereço não pode ser atribuído a um *host* específico. A Figura 19.17 mostra três exemplos de endereços de rede, um em cada classe.

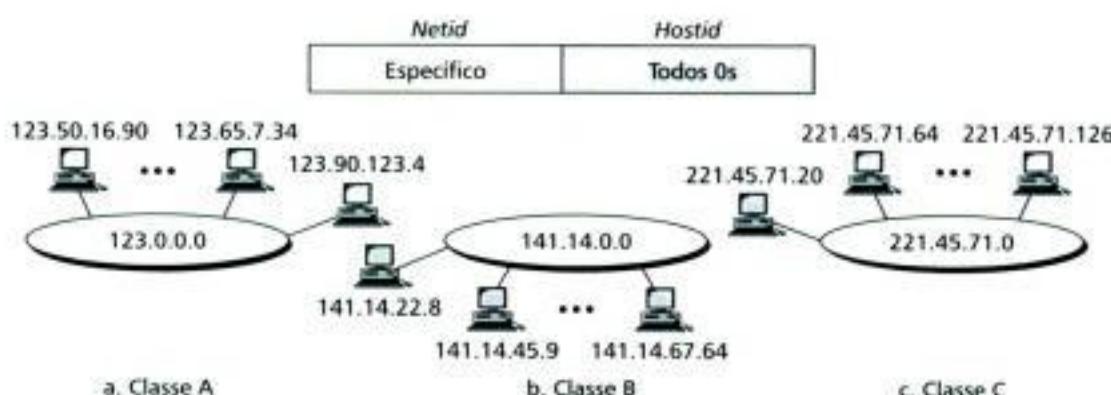


Figura 19.17 Endereço de rede.

Os endereços das redes desempenham um papel fundamental dentro das classes de endereçamento. Um endereço de rede possui algumas propriedades:

1. Todos os *bytes* de *hostid* são 0.
2. O endereço da rede define a rede particular para o restante da Internet. Mais tarde veremos como os roteadores conseguem rotear pacotes baseados nesses endereços.
3. O endereço da rede é o primeiro endereço da faixa.
4. Dado o endereço da rede, podemos determinar a classe de endereçamento.

Nas classes de endereçamento, o endereço da rede, é atribuído a uma empresa ou organização por um órgão competente*.

Exemplo 5

Dado o endereço 23.56.7.91, determine o endereço da rede.

Solução

Este endereço pertence à classe A. Um único *byte* define a parte de rede (*netid*). Podemos determinar o endereço da rede substituindo os *bytes* de *hostid* (56.7.91) por 0s. Assim, o endereço dessa rede é 23.0.0.0.

Exemplo 6

Dado o endereço 132.6.17.85, determine o endereço da rede.

Solução

Este endereço pertence à classe B. Os dois primeiros *bytes* definem a parte de rede (*netid*). Podemos determinar o endereço da rede substituindo os *bytes* de *hostid* (17.85) por 0s. Assim, o endereço dessa rede é 132.6.0.0.

Exemplo 7

Dado o endereço da rede 17.0.0.0, determine a classe de endereçamento.

Solução

Este endereço de rede pertence à classe A, pois a parte de rede (*netid*) tem 1-byte somente.

Um endereço da rede é diferente de um *netid*. O endereço da rede possui tanto a parte de *netid* quanto a parte de *hostid*. Esta última composta totalmente de 0s.

Um exemplo de Internet: Classes de Endereçamento

A Figura 19.18 mostra parte de uma internet, formada por cinco redes.

1. Uma LAN Token Ring cujo endereço da rede é 220.3.6.0 (classe C).
2. Uma LAN Ethernet cujo endereço da rede é 134.18.0.0 (classe B).
3. Uma LAN Ethernet cujo endereço da rede é 124.0.0.0 (classe A).
4. Uma WAN ponto a ponto (linha tracejada). Esta rede (por exemplo, uma linha T-1 ou E-1) fecha o enlace entre dois roteadores; não há outros *hosts* senão as interfaces dos roteadores. Neste caso, para economizar endereços, nenhum endereço da rede é atribuído para este tipo de WAN. Uma WAN comutada (tal como Frame Relay ou ATM) pode ser conectada através de muitos roteadores. Mostramos três na figura. Um roteador permite o acesso à WAN para a rede Token Ring. Além disso, as redes Ethernet são conectadas aos

* N. de R. T.: No Brasil, a obtenção de endereços IP é feita junto à FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo).

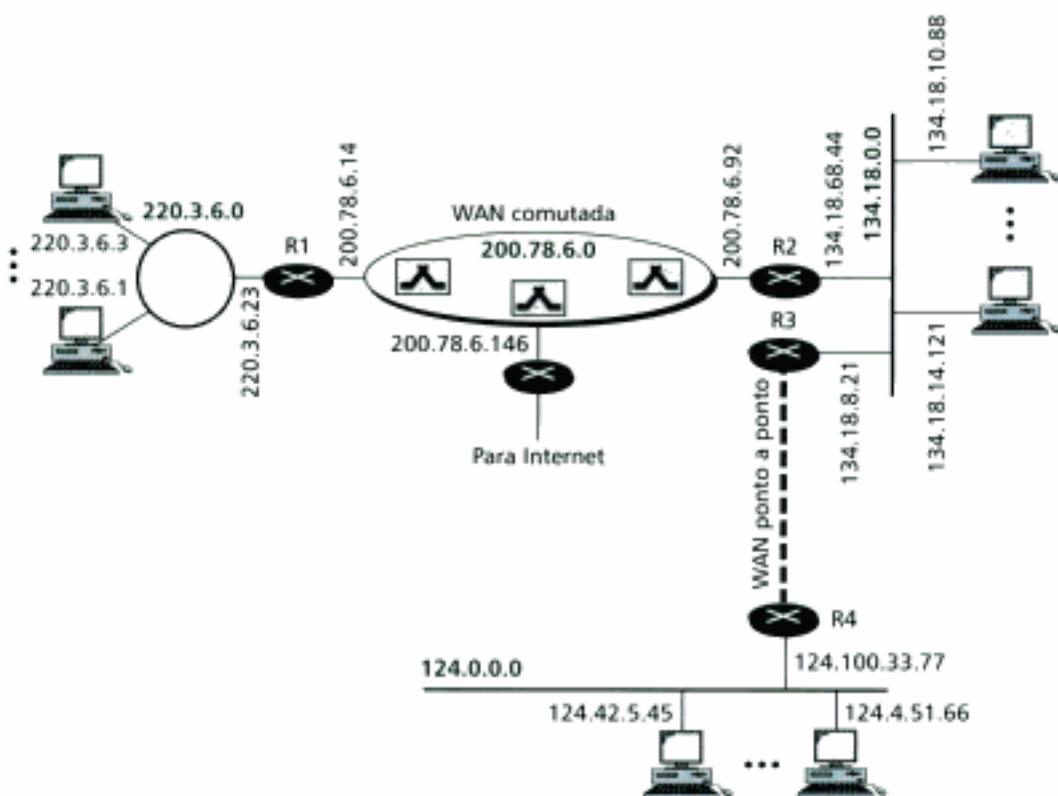


Figura 19.18 Exemplo de internet.

respectivos roteadores que as interligam a um outro roteador, através dos nós de comunicação, para conexão dessa WAN ao restante da Internet.

Sub-redes

Quando uma empresa ou organização recebe um endereço classe A, B ou C, o primeiro endereço na faixa define o endereço da rede. Este endereço é utilizado pelos roteadores externos à empresa, como veremos, para rotear os pacotes destinados à rede dessa empresa. O mundo externo reconhece apenas esta rede e não um endereço individual dentro dela.

Uma parte do endereço de 32-bits identifica a rede (*netid*) e a outra parte identifica o *host* (*hostid*) na rede. Isto significa que há um senso de hierarquia no esquema de endereçamento IP. Para chegar até um *host* particular na Internet devemos primeiramente chegar até a rede, usando a primeira parte do endereço (*netid*). Encontrando a rede podemos alcançar o *host* particular usando a segunda parte do endereço (*hostid*). Logo, os endereços IP foram criados com dois níveis de hierarquia. A Figura 19.19 mostra o conceito.

Os endereços IP foram criados com dois níveis de hierarquia.

Contudo, freqüentemente um administrador da rede de uma empresa precisa juntar os *hosts* dentro de grupos. Neste caso, a rede precisa ser dividida em muitas **sub-redes (subnets)**. Por exemplo, uma universidade pode desejar agrupar os *hosts* de acordo com os departamentos. Sendo assim, a universidade continuará possuindo apenas um endereço da rede, mas precisará de muitos endereços de sub-redes. O mundo externo continuará enxergando a universidade através do endereço da rede original. Voltando ao caso da empresa, internamente à empresa cada sub-rede é reorganizada através dos endereços de sub-rede. No processo de **criação das sub-redes (subnetting)** uma rede é dividida em muitos grupos menores, onde cada sub-rede terá um número de endereço próprio dela.

Hidden page

Hidden page

Hidden page

A quantidade de sub-redes criadas é determinada através do número de *bits* em 1 tomados emprestados da parte de *hostid*. Se o número de *bits* 1s extras for n , a quantidade de sub-redes criadas será 2^n . Se N representa a quantidade de sub-redes, o número de *bits* 1s extras a ser tomado emprestado da parte de *hostid* é $\log_2 N$.

Exemplo 9

Um roteador dentro da empresa recebe outro pacote com endereço de destino 190.240.33.91. Mostre como o roteador determina o endereço da sub-rede para rotear o pacote.

Solução

O roteador executa três passos:

- O roteador conhece a máscara. Ele assume que a máscara é /19, conforme ilustra a Figura 19.23.
- O roteador aplica a máscara ao endereço (190.240.33.91). O endereço da sub-rede é 190.240.32.0.
- O roteador procura na tabela de roteamento determinar caminhos para rotear o pacote para este destino particular. Mais tarde veremos o que acontece se este destino não existir.

Supernetting

Embora as classes de endereços A e B estejam praticamente exauridas, os endereços de classe C ainda estão disponíveis. Entretanto, a quantidade de *hosts* numa faixa de endereços de uma rede classe C (256 endereços) pode não ser satisfatória para algumas empresas. Até mesmo uma empresa de porte médio pode precisar de mais endereços.

Uma solução é usar o **supernetting**. Utilizando o conceito de supernetting uma empresa pode combinar muitas faixas de endereços classe C para criar uma faixa maior de endereços. Assim, essas redes são combinadas para criar uma *supernetwork*. Para isto, uma empresa pode trabalhar com um conjunto de faixas de endereços classe C, em vez de apenas uma. Por exemplo, uma empresa que necessitar de 1000 endereços IP válidos pode adquirir quatro faixas de endereços classe C. Desse modo, esta empresa pode montar uma *supernetwork*. Existem muitas questões ligadas à criação de supernetting que não serão discutidas aqui, pois fogem ao escopo deste livro. Para mais informações, veja Forouzan, *TCP/IP Protocol Suite*, 2d ed., McGraw-Hill, 2002.

Endereçamento sem Classes

A idéia das classes de endereçamento pareceu interessante e fácil de gerenciar num primeiro instante, mas ela criou muitos problemas. Até meados da década de 90, uma faixa de endereços significava um grupo de endereços nas classes A, B ou C. O número mínimo de endereços concedidos a uma empresa ou organização era 256 (classe C) e o número máximo era 16.777.216 (classe A). Entre estes limites uma empresa poderia escolher uma faixa classe B ou muitas faixas classe C e usar, por exemplo, supernetting. Entretanto, as escolhas eram limitadas. Além disso, e as pequenas empresas que necessitavam de apenas 16 endereços? E os usuários domésticos que necessitavam de somente um ou dois endereços?

Durante a década de 90, os provedores de acesso à Internet (ISPs) ganharam notoriedade. Um ISP é uma empresa que provê acesso à Internet a indivíduos, empresas pequenas e médias que não querem registrar um domínio na Internet e prover serviços de acesso à Internet (como serviços de *e-mail*) aos funcionários. Um ISP pode prover estes serviços para estas empresas. Um ISP pode adquirir muitas faixas classe B ou C e, então, subdividi-las em grupos de 2, 4, 8 ou 16 endereços para acesso domiciliar e de pequenas empresas. Os consumidores podem se conectar aos ISPs através de *modem* de linha discada, *modem* DSL ou *cable modem*. Contudo, cada usuário ainda necessita de um endereço IP.

Em 1996, o comitê de padronização da Internet anunciou uma nova arquitetura denominada **endereçamento sem classes** que tornaria as classes de endereçamento obsoletas.

Faixas de Tamanhos Variáveis

A idéia central do endereçamento sem classes é dispor faixas de endereços IP de tamanho variável pertencentes a nenhuma classe particular de endereços. Nesse caso, podemos ter faixa de endereços com 2, 4, 8, 128, ... endereços. Há algumas restrições, como veremos, mas em geral nesse esquema uma faixa de endereços pode ser muito pequena ou muito grande. Nesta arquitetura, toda a faixa de endereços IP (2^{32} endereços) foi dividida em faixas de tamanhos diferentes. Nela, uma empresa pode solicitar a quantidade de endereços IP ideal para o número de *hosts* dela. Há uma única restrição quanto ao número de endereços solicitados, para que sigam a potência de 2 (2, 4, 8, ...). Assim, um usuário doméstico pode solicitar 2 endereços. Uma pequena empresa pode precisar e solicitar apenas 16 endereços. Uma empresa de médio porte pode solicitar 1024 endereços e assim por diante. Note que este esquema é muito mais flexível que as classes de endereçamento.

Máscara

Você deve estar lembrado que quando uma empresa recebe um endereço dentro de uma classe, a ela foi concedido o endereço da rede de identificação e, por conseguinte, da máscara padrão. No processo de criação das sub-redes, a empresa recebe o endereço da rede e o administrador subdivide a rede em sub-redes, gerando a máscara personalizada. O endereçamento sem classes segue o mesmo conceito. Quando uma empresa ou organização solicita um endereço IP, ela recebe o primeiro endereço e a máscara. Estas duas partes da informação definem toda a faixa de IPs disponível para ela. Normalmente, a máscara é fornecida na notação de barra, conforme discutimos antes.

Determinando o Endereço da Rede

Podemos determinar o endereço da rede (a identificação de uma empresa ou organização dentro de uma *internetworking*) a partir de um endereço da faixa e da máscara de sub-rede? A resposta é definitivamente sim. Podemos aplicar a máscara ao endereço da faixa (através do processo de ANDing) e descobrir o endereço da rede.

Criação de Sub-redes

No esquema de endereçamento sem classes é claro que podemos utilizar o conceito de sub-redes. Quando uma empresa ou organização recebe um endereço IP, ela pode criar sub-redes de acordo com as necessidades dela. O administrador da rede pode calcular a máscara de sub-rede da mesma maneira que discutimos para as classes de endereçamento. Aliás, o procedimento é ainda mais simples aqui. A quantidade de 1s na máscara (n) aumenta para definir a máscara de sub-rede. Por exemplo, se a máscara padrão é /17, a máscara personalizada pode ser /20 para criar oito sub-redes ($2^3 = 8$).

CIDR

A idéia central que permeia o esquema de endereçamento sem classes é o **Classless InterDomain Routing (CIDR)**. Embora o endereçamento sem classes diminua o desperdício de endereços IP, precisamos de um novo esquema de roteamento para o endereçamento sem classes ou CIDR. O CIDR promove o roteamento de pacotes nas redes sem classes de endereçamento.

Configuração Dinâmica de Endereços

Cada computador que se conecta à Internet deve carregar as seguintes informações:

- Um endereço IP próprio
- Uma máscara de sub-rede

Hidden page

Hidden page

ceber IP individual para cada *host*. Com a redução na quantidade de endereços IPs disponíveis nos órgãos responsáveis pela distribuição criou-se um problema sério.

Uma solução rápida para este problema é denominada **Network Address Translation (NAT)**. O recurso NAT habilita um usuário a utilizar quantos endereços internos ele quiser, a partir de um ou de um conjunto pequeno de endereços externos. Observe a distinção entre endereços internos e externos. Os endereços internos gerados pelo NAT são conhecidos apenas localmente, isto é, dentro da rede em questão. Os endereços externos são recebidos dos órgãos que administram a Internet e, por isso, são conhecidos globalmente. Assim, o tráfego interno pode utilizar um conjunto enorme de endereços, enquanto o tráfego externo utiliza um conjunto pequeno de endereços IP.

Para separar os endereços usados dentro de empresas e/ou residências dos endereços globais utilizados na Internet, os órgãos que gerenciam a Internet reservaram três conjuntos de endereços como privativos, mostrados na Tabela 19.2.

TABELA 19.2 Endereços para redes privadas

Faixa	Total
10.0.0.0	2^{24}
172.16.0.0	2^{20}
192.168.0.0	2^{16}

Qualquer organização pode utilizar um destes três tipos de endereços sem o conhecimento prévio dos órgãos competentes. É domínio público que estes endereços reservados são de uso exclusivo em redes privadas. Estes endereços são únicos dentro da empresa ou organização, mas não são únicos globalmente. Nenhum roteador encaminhará pacotes a um deles, quando utilizados como endereços de destino.

Para implementar este recurso, um *site* deve possuir uma única conexão com a Internet global através de roteador rodando o *software* com suporte NAT. A Figura 19.25 ilustra uma implementação simples do NAT.

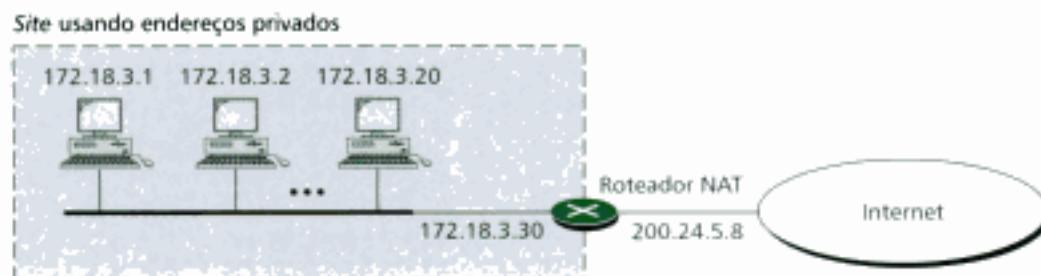


Figura 19.25 NAT.

De acordo com a figura, a rede privada utiliza endereços privados. O roteador que conecta esta rede à rede global utiliza um endereço privado e um endereço global. A rede privada é transparente para o resto da Internet, pois o roteador NAT tem um endereço válido (200.24.5.8) que é enxergado pela rede global.

Tradução de Endereços

Todos os pacotes direcionados para fora do roteador NAT têm o *endereço de origem* do pacote interno substituído pelo endereço externo (global) no ato da passagem pelo roteador. Da mesma forma, todos os pacotes direcionados para dentro da rede do roteador NAT têm o campo de *endereço de destino* do pacote externo substituído pelo endereço privado apropriado. A Figura 19.26 mostra um exemplo de tradução de endereços.

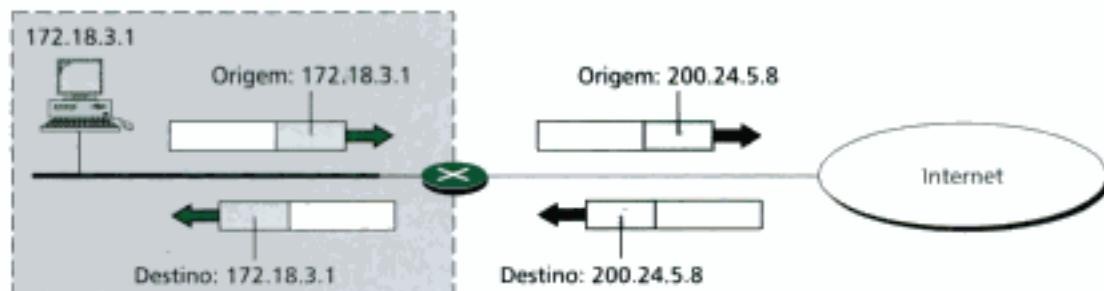


Figura 19.26 Tradução de endereços.

Tabela de Tradução

O leitor deve ter percebido que a tradução do endereço interno do pacote para o endereço externo é direta. Mas, como o roteador NAT sabe que endereço de destino do pacote deve atribuir para um pacote chegando da Internet? Numa rede grande, há possivelmente milhares ou centenas de endereços IP privados para um *host* específico. O problema é resolvido se o roteador NAT tiver uma tabela de tradução.

Usando um Endereço IP Na sua forma mais simples, uma tabela de tradução possui somente duas colunas: endereços privados e endereços externos (endereço de destino do pacote). Quando um roteador traduz o endereço de origem do pacote de saída ele também registra o endereço de destino, isto é, o endereço para onde o pacote seguiu. Quando uma resposta chega do destino, o roteador utiliza o endereço de origem do pacote para determinar o endereço privado do pacote. A Figura 19.27 ilustra a idéia. Perceba que os endereços modificados (traduzidos) são mostrados em cada situação.

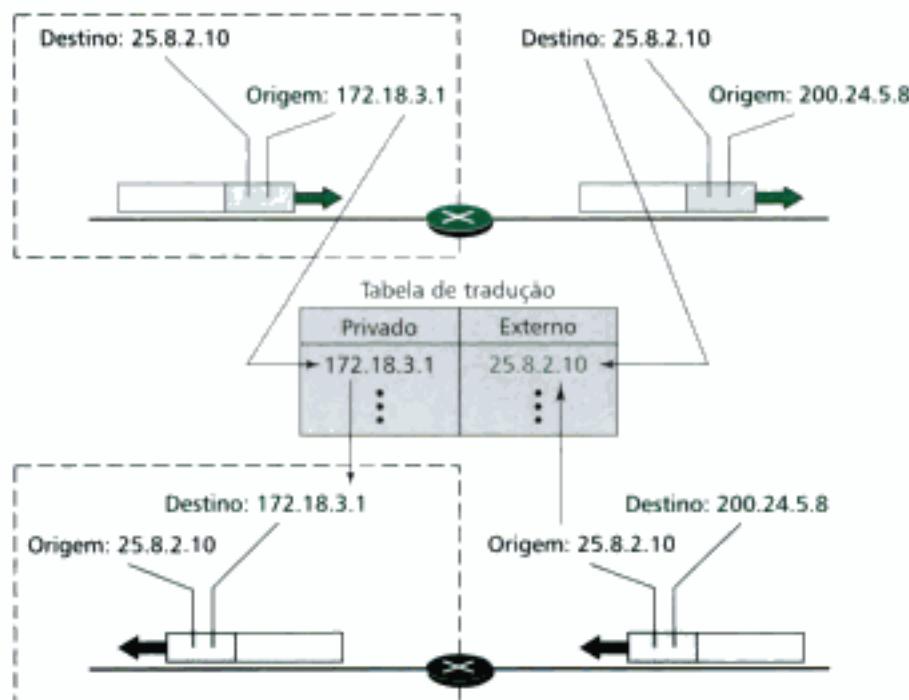


Figura 19.27 Tradução.

Nesta estratégia, uma comunicação deve sempre ser iniciada pela rede privada. A descrição do mecanismo NAT requer que a rede privada inicie a comunicação. Como veremos, a solução NAT é bastante utilizada pelos ISPs, os quais atribuem um único endereço IP aos usuários. Um usuário, entretanto, pode ser membro de uma rede privada possuindo uma enorme quantidade de endereços privados. Neste caso, a comunicação com a Internet sempre é iniciada pelo usuário, usando um programa cliente tal como HTTP, TELNET ou FTP para acessar o programa servidor corresponden-

te. Por exemplo, quando um *e-mail* originado fora da zona de abrangência NAT é recebido pelo servidor de *e-mails* do ISP, o *e-mail* permanece armazenado na caixa de correio até que o usuário resolva baixá-lo. Uma rede privada não pode rodar um programa servidor para clientes fora da rede se ela estiver usando a tecnologia NAT.

Usando um Pool de Endereços IP Visto que o roteador NAT possui apenas um endereço global, apenas um *host* da rede privada pode acessar o mesmo *host* externo. Para remover esta restrição, o roteador NAT usa um *pool* de endereços globais. Por exemplo, ao invés de usar somente um endereço global (200.24.5.8), o roteador NAT pode usar quatro endereços (200.24.5.8, 200.24.5.9, 200.24.5.10 e 200.24.5.11). Nesse caso, quatro *hosts* da rede privada podem estabelecer comunicação com o mesmo *host* externo, ao mesmo tempo, porque cada par de endereços define uma conexão. Entretanto, ainda existem algumas desvantagens nessa solução. Não mais que quatro conexões podem ser estabelecidas com o mesmo destino. Nenhum *host* da rede privada pode acessar dois servidores de programas externos (por exemplo, HTTP e FTP) ao mesmo tempo.

Usando Tanto Endereço IP Quanto Números de Portas Para permitir o relacionamento entre muitos *hosts* da rede interna e muitos servidores de programas localizados fora dela, precisamos adicionar mais informação na tabela de tradução. Por exemplo, suponha que dois *hosts* dentro de uma rede privada com endereços 172.18.3.1 e 172.18.3.2 necessitem acessar um servidor HTTP hospedado no *host* externo 25.8.3.2. Se a tabela de tradução tiver cinco colunas, em vez de duas, incluindo o número da porta do protocolo da camada de transporte que fez a chamada (origem) e o número da porta que atendeu à chamada (destino), a ambigüidade é totalmente eliminada.

TABELA 19.3 Tabela de tradução de cinco colunas

Endereço privado	Porta privada	Endereço externo	Porta externa	Protocolo de transporte
172.18.3.1	1400	25.8.3.2	80	TCP
172.18.3.2	1401	25.8.3.2	80	TCP
172.18.3.1	1402	25.8.3.2	80	UDP
...

Perceba que quando chega a resposta do servidor HTTP, a combinação endereço de origem (25.8.3.2) e número da porta de destino (1400) define univocamente o *host* na rede privada para o qual a comunicação deve ser direcionada. Note ainda que, para o perfeito funcionamento desta tradução, os números das portas temporárias (1400, 1401 e 1402) devem ser únicos.

19.3 ROTEAMENTO

Conforme discutido na Seção 19.2 precisamos de endereços e de roteamento para controlar o processo de entrega de pacotes. Na seção passada discutimos o endereçamento, nosso foco agora é o roteamento.

Técnicas de Roteamento

Para que ocorra roteamento numa rede é necessário que um *host* ou roteador tenha uma **tabela de roteamento**. Quando um *host* tem um pacote a ser enviado ou quando um roteador recebeu um pacote externo, o *host* que realiza o roteamento sempre procura na tabela de roteamento a rota para o destino final. Contudo, hoje em dia esta solução simples é impossível numa *internetwork*, tal como a Internet, porque a quantidade de entradas na tabela de roteamento torna a tabela de pesquisa (*lookups*) ineficiente. Muitas técnicas podem tornar o tamanho da tabela de roteamento gerenciável de modo a permitir o controle de questões essenciais nas redes, tal como a segurança. Veremos alguns destes métodos nesta seção.

Hidden page

bora não seja eficiente adicionar endereços de *hosts* individuais numa tabela de roteamento, existem algumas situações onde o administrador da rede quer ter o controle sobre o roteamento. Por exemplo, na Figura 19.30, se o administrador quiser que todos os pacotes destinados ao *host* B sejam entregues via roteador R3, em vez de R1, uma única entrada na tabela de roteamento do *host* A pode definir, explicitamente, a rota.

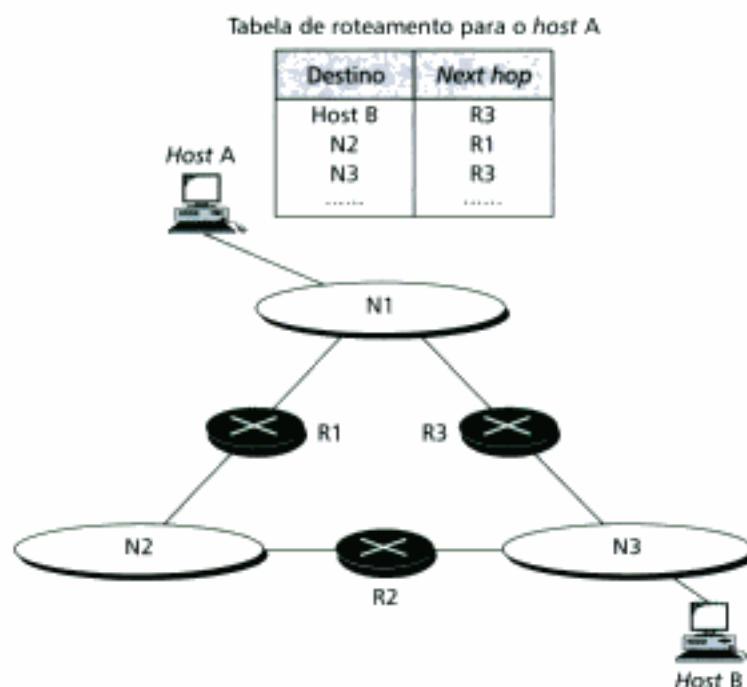


Figura 19.30 Roteamento para *host* específico.

Rota Padrão

Outra técnica que simplifica o roteamento é a *rota padrão*. Na Figura 19.31, um *host* A é conectado a uma rede com dois roteadores. O roteador R1 é utilizado para rotear os pacotes dos *hosts* conectados à rede N2. Entretanto, para o restante da Internet, o roteador R2 é usado. Assim, em vez de listar todas as redes da Internet, o *host* A pode ter apenas uma entrada ou *rota padrão* ou *default* (endereço de rede 0.0.0.0).

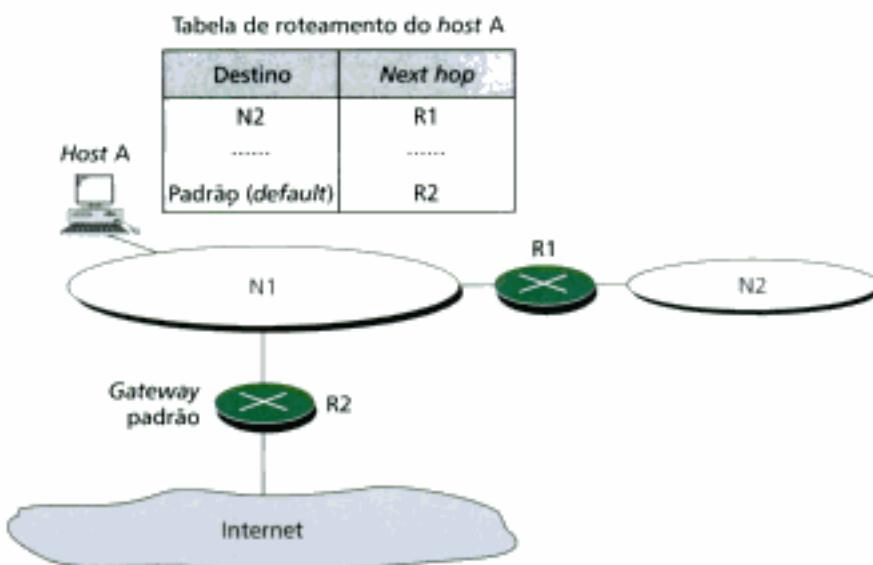


Figura 19.31 Rota padrão.

Roteamento Estático versus Roteamento Dinâmico

Um roteador e/ou *host* mantém uma tabela de roteamento com uma entrada para cada destino de modo a rotear os pacotes IP. A tabela pode ser de roteamento estático ou dinâmico.

Tabela de Roteamento Estático

Uma **tabela de roteamento estático** armazena informação de rotas digitadas manualmente. O administrador da rede entra com a rota para cada destino na tabela. Quando este tipo de tabela é criada, ela não pode ser atualizada automaticamente, por exemplo, quando houver alguma mudança nas rotas da Internet. A tabela deve ser alterada manualmente pelo administrador.

Uma tabela de roteamento estático pode ser utilizada numa internet pequena, sem previsões de mudanças a curto prazo na estrutura da rede, ou numa internet experimental para diagnóstico de erros. Em geral, não é uma boa estratégia utilizar roteamento estático numa internet grande, tal como a Internet.

Tabela de Roteamento Dinâmico

Uma **tabela de roteamento dinâmico** armazena informação de rotas atualizadas automaticamente em períodos de tempos regulares. Para tanto, o dispositivo de roteamento utiliza protocolos de roteamento dinâmico tal como o RIP, OSPF ou BGP (veja Capítulo 21). Sempre que ocorrer uma modificação na estrutura da rede, tal como o desligamento de um roteador ou quedas de *links*, os protocolos de roteamento dinâmicos atualizam todas as tabelas nos roteadores (e eventualmente nos *hosts*).

Os roteadores numa internet grande, tal como a Internet, precisam ser atualizados dinamicamente para aumentar a eficiência do processo de entrega de pacotes IP. Estudaremos em detalhes os protocolos de roteamento dinâmico no Capítulo 21.

Tabela de Roteamento para as Classes de Endereçamento

No endereçamento baseado em classes, com ou sem sub-redes, uma tabela de roteamento precisa de no mínimo quatro colunas (normalmente ela tem mais): máscara, endereço da rede de destino, endereço do próximo salto e interface, conforme ilustra a Figura 19.32.

	Máscara	Endereço de destino	Endereço do próximo salto (next-hop)	Interface
Host-específico	/8	14.0.0.0	118.45.23.8	m1
	/32	192.16.7.1	202.45.9.3	m0
	/24	193.14.5.0	84.78.4.12	m2
Padrão (default)	/0	/0	145.11.10.6	m0

Figura 19.32 Tabela de roteamento para endereçamento baseado em classes.

Quando um pacote chega ao roteador, ele aplica a máscara ao **endereço de destino** para determinar o endereço correspondente do destino. Caso encontre, o pacote é enviado para a rede através da interface correspondente na tabela. Caso não encontre o endereço da rede de destino, o pacote é entregue à interface padrão que transporta o pacote para o roteador padrão (*default*).

Exemplo 10

Usando a tabela da Figura 19.32, o roteador recebe um pacote destinado ao endereço 192.16.7.1. A máscara é aplicada ao endereço de destino em cada linha até que um destino seja encontrado. Neste exemplo, o roteador envia o pacote através da interface m0 (*host* específico).

Exemplo 11

Usando a tabela da Figura 19.32, o roteador recebe um pacote destinado ao endereço 193.14.5.22. A máscara é aplicada ao endereço de destino em cada linha até que o endereço do próximo salto

correspondente seja determinado. Neste exemplo, o roteador envia o pacote através da interface m2 (rede específica).

Exemplo 12

Usando a tabela da Figura 19.32, o roteador recebe um pacote destinado ao endereço 200.34.12.34. A máscara é aplicada ao endereço de destino em cada linha, mas nenhuma correspondência é determinada. Neste exemplo, o roteador envia o pacote através da interface padrão m0.

Tabela de Roteamento para Endereçamento sem Classes

Por enquanto, a discussão sobre tabelas de roteamento ficou concentrada no esquema baseado nas classes de endereçamento. Agora precisamos considerar o endereçamento sem classes e o Classless InterDomain Routing (CIDR). A troca para o endereçamento sem classes requer mudanças na organização das tabelas e nos algoritmos de roteamento.

Tamanho da Tabela de Roteamento

Quando usamos o esquema de endereçamento baseado em classes, há uma única entrada na tabela de roteamento para cada *site* fora da rede. A entrada define o *site* até mesmo se ele estiver dividido em sub-redes. Quando um pacote chega ao roteador da rede de destino, verifica a entrada correspondente e direciona o pacote para a interface correta.

Quando usamos o esquema de endereçamento sem classes, a quantidade de entradas na tabela do roteador pode aumentar ou diminuir. Ela diminuirá se a faixa de endereços atribuídos a uma empresa ou organização for maior que a faixa original no esquema baseado em classes. Por exemplo, ao invés de quatro entradas para uma organização que cria uma *supernet* a partir de quatro endereços classe C, podemos ter apenas uma entrada na tabela de roteamento sem classes.

É mais provável, entretanto, que o número de entradas da tabela de roteamento aumente. Isso é assim porque a intenção do esquema de endereçamento sem classes é dividir as faixas de endereços das classes A e B. Por exemplo, em vez de atribuir 16 milhões de endereços a uma única organização, os endereços podem ser distribuídos entre muitas organizações. O problema é que, considerando que havia apenas uma entrada na tabela de roteamento para um endereço classe A, passamos a ter muitas entradas nas tabelas utilizadas no endereçamento sem classes. Assim, por exemplo, se um endereço classe B (cerca de 64.000 endereços) é dividido entre 60 empresas ou organizações, há 60 entradas na tabela de roteamento onde antes havia apenas uma.

Roteamento Hierárquico

Para resolver o problema gigantesco provocado pelas tabelas de roteamento foi criado um senso de hierarquia na arquitetura da Internet, originando assim, tabelas de **roteamento hierárquico**. No Capítulo 1, mencionamos que a Internet hoje carrega um forte senso de hierarquia. Vimos que a Internet é dividida em ISPs nacionais e internacionais. Os ISPs nacionais são subdivididos em ISPs regionais, que são subdivididos em ISPs locais. Se a tabela de roteamento tiver um senso de hierarquia como a arquitetura da Internet tem, o tamanho da tabela de roteamento pode ser diminuído drasticamente.

Vamos olhar o caso de um ISP local. Um ISP local pode possuir uma ou mais faixas de endereços com uma certa máscara. O ISP local pode subdividir uma faixa em faixas menores e de tamanhos diferentes para atribuí-las aos usuários individuais ou às empresas, grandes ou pequenas. Se a faixa de endereços atribuída ao ISP local é A.B.C.D/n, este ISP pode criar blocos de E.F.G.H/m, onde *m* pode variar para cada usuário, com *m* > *n*.

Isto reduz o tamanho da tabela de roteamento? O restante da Internet não toma conhecimento desta divisão. Todos os usuários do ISP local estão definidos como A.B.C.D/n para o resto da Internet. Logo, todo pacote destinado a um dos endereços localizados nesta faixa é roteado pelo ISP local. Assim, há uma única entrada em cada roteador do mundo para todos estes usuários. Eles pertencem ao mesmo grupo. Claro, dentro do ISP local o roteador deve reconhecer as sub-redes de modo a rotear pacotes aos destinos corretos. Se um usuário dentro destes grupos for uma empresa grande, ela poderá criar outro nível de hierarquia subdividindo em sub-redes menores o endereço original. No roteamento sem classes, os níveis hierárquicos são ilimitados uma vez que seguimos as regras do endereçamento sem classes.

Roteamento Geográfico

Outra forma de diminuir o tamanho da tabela de roteamento é estender globalmente a idéia do roteamento hierárquico de modo a criar um **roteamento geográfico**. A idéia é dividir as faixas de endereços em endereços menores. Podemos atribuir faixas para as Américas, Europa, Ásia, África e assim por diante. Os roteadores das ISPs fora da Europa teriam apenas uma ou poucas entradas nas tabelas de roteamento para pacotes direcionados ao continente europeu. Do mesmo modo, os roteadores fora das Américas teriam somente uma ou poucas entradas nas tabelas de roteamento para pacotes direcionados para algum lugar nas Américas. Parte desta idéia foi implementada de fato para o endereçamento classe C. Mas, para eficiência total, as classes A e B precisariam ser recicladas e reatribuídas.

Algoritmos de Busca das Tabelas de Roteamento

No esquema de endereçamento sem classes, os mecanismos de busca são definitivamente mais complexos. Há necessidade de modificar os **algoritmos de busca** para tornar o CIDR tão eficiente quanto possível. Muitos algoritmos novos têm sido propostos para tal finalidade, mas não os estudaremos pois fogem ao escopo deste livro.

19.4 TERMOS-CHAVE

Abordagem de circuitos virtuais (para comutação de pacotes)	Máscara
Abordagem de datagrama (para comutação de pacotes)	Máscara de sub-rede
Algoritmo de busca	Máscara padrão
Classes de endereçamento	<i>Netid</i>
Classless InterDomain Routing (CIDR)	Network Address Translation (NAT)
Comutação de pacotes	Notação binária
Criação de sub-redes (<i>subnetting</i>)	Notação de barra
Datagrama	Notação decimal com pontos
Dynamic Host Configuration Protocol (DHCP)	Rede de comutação de pacotes
Endereçamento sem classes	Rota padrão
Endereço classe A	Roteamento geográfico
Endereço classe B	Roteamento hierárquico
Endereço classe C	Roteamento <i>next-hop</i> (roteamento do próximo salto)
Endereço classe D	Roteamento para <i>host</i> específico
Endereço classe E	Roteamento para rede específica
Endereço da rede	Serviços orientados à conexão
Endereço de Internet	Serviços sem conexão
Endereço IP	Sub-redes (<i>subnet</i>)
Endereço <i>multicast</i>	<i>Supernetting</i>
Endereço <i>unicast</i>	Tabela de roteamento
Faixa de endereços	Tabela de roteamento dinâmico
<i>Hostid</i>	Tabela de roteamento estático

19.5 RESUMO

- Existem duas abordagens comuns para a comutação de pacotes: comutação de datagramas e comutação de circuitos virtuais.
- Na comutação de datagramas cada pacote é tratado independentemente dos demais.
- O sistema de endereçamento global da camada de rede identifica univocamente cada *host* (incluindo roteadores) para o processo de entrega de pacotes através das redes.
- O endereço de Internet ou endereço IP tem 32-bits de tamanho (IPv4), definindo univoca e universalmente *hosts* (incluindo roteadores) na Internet.
- A parte do endereço IP que identifica a rede é denominada *netid*.
- A parte do endereço IP que identifica os *hosts* na rede, incluindo roteadores, é denominada *hostid*.

Hidden page

Hidden page

31. Uma máscara de sub-rede de classe C tem vinte e cinco 1s. Quantas sub-redes esta máscara define?
- 2
 - 8
 - 16
 - 0
32. Dado o endereço IP 201.14.78.65 e a máscara de sub-rede 255.255.255.224 (/27), qual é o endereço da sub-rede a qual pertence esse endereço IP?
- 201.14.78.32
 - 201.14.78.65
 - 201.14.78.64
 - 201.14.78.12
33. Dado o endereço IP 180.25.21.172 e a máscara de sub-rede 255.255.192.0 (/18), qual é o endereço da sub-rede a qual pertence esse endereço IP?
- 180.25.21.0
 - 180.25.0.0
 - 180.25.8.0
 - 180.0.0.0
34. Dado o endereço IP 18.250.31.14 e a máscara de sub-rede 255.240.0.0 (/18), qual é o endereço da sub-rede a qual pertence esse endereço IP?
- 18.0.0.14
 - 18.31.0.14
 - 180.240.0.0
 - 18.9.0.14
35. A classe _____ possui o maior número de hosts por endereço da rede.
- A
 - B
 - C
 - D
36. _____ é um programa cliente-servidor que provê um endereço IP, máscara de sub-rede, endereço da interface de um roteador e o endereço IP do servidor de nomes a um computador.
- NAT
 - CIDR
 - ISP
 - DHCP
37. Sobre uma rede que usa o NAT, o _____ possui uma tabela de tradução.
- Switch
 - Roteador
 - Servidor
 - Nenhuma das alternativas anteriores
38. Sobre uma rede que usa o NAT, o _____ inicia o processo de comunicação.
- Um host externo
 - Um host interno
 - O roteador
 - (a) ou (b)
39. Sobre uma rede que usa o NAT, o roteador pode usar _____ de endereço(s) global(is).
- Um
 - Dois
 - Um pool de
 - Nenhuma das alternativas anteriores
40. No roteamento _____, o endereço IP completo de um destino é colocado na tabela de roteamento.
- Next-hop* (próximo salto)
 - Para rede específica
 - Para host específico
 - Por rota padrão
41. No roteamento _____, tanto a máscara quanto o endereço de destino são 0.0.0.0 na tabela de roteamento.
- Next-hop* (próximo salto)
 - Para rede específica
 - Para host específico
 - Por rota padrão
42. No roteamento _____, o endereço de destino é um endereço de rede na tabela de roteamento.
- Next-hop* (próximo salto)
 - Para rede específica
 - Para host específico
 - Por rota padrão

Exercícios

43. Complete a Tabela 19.4 para estabelecer uma comparação entre as redes de comutação de circuitos e as redes de comutação de pacotes.
44. Complete a Tabela 19.5 para estabelecer uma comparação entre a abordagem de datagramas e a abordagem de
- circuitos virtuais para uma rede de comutação de pacotes.
45. Converta os endereços IP abaixo para a notação binária.
- 114.34.2.8
 - 129.14.6.8
 - 208.34.54.12

TABELA 19.4 Exercício 43

Característica	Comutação de circuitos	Comutação de pacotes
Caminho dedicado		
<i>Store and Forward</i>		
Necessidade de estabelecimento da conexão		
Tabela de roteamento		

TABELA 19.5 Exercício 44

Característica	Datagrama	Círculo virtual
Todos os pacotes seguem a mesma rota		
Tabela de pesquisa (<i>lookup</i>)		
Estabelecimento da conexão		
Ordem de chegada dos pacotes		

- d. 238.34.2.1
e. 241.34.2.8
46. Converta os endereços IP abaixo para a notação decimal com pontos.
- a. 01111111 11110000
01100111 01111101
- b. 10101111 11000000
11111000 00011101
- c. 11011111 10110000
00011111 01011101
- d. 11101111 11110111
11000111 00011101
- e. 11110111 11110011
10000111 11011101
47. Determine a classe de endereçamento dos seguintes endereços IP:
- a. 208.34.54.12
b. 238.34.2.1
c. 114.34.2.8
d. 129.14.6.8
e. 241.34.2.8
48. Determine a classe de endereçamento dos seguintes endereços IP:
- a. 11110111 11110011
10000111 11011101
- b. 10101111 11000000
11110000 00011101
- c. 11011111 10110000
00011111 01011101
- d. 11101111 11110111
11000111 00011101
- e. 01111111 11110000
01100111 01111101
49. Determine o *netid* e o *hostid* dos seguintes endereços IP:
- a. 114.34.2.8
b. 19.34.21.5
50. Determine o *netid* e o *hostid* dos seguintes endereços IP:
- a. 129.14.6.8
b. 132.56.8.6
c. 171.34.14.8
d. 190.12.67.9
51. Determine o *netid* e o *hostid* dos seguintes endereços IP:
- a. 192.8.56.2
b. 220.34.54.12
c. 208.34.54.12
d. 205.23.67.8
52. Numa sub-rede classe A sabemos que o endereço de um *host* e da máscara são:
- Endereço IP: 25.34.12.56
Máscara: 255.255.0.0
- Qual é o endereço dessa sub-rede?
53. Numa sub-rede classe B sabemos que o endereço de um *host* e da máscara são:
- Endereço IP: 125.134.112.66
Máscara: 255.255.224.0
- Qual é o endereço dessa sub-rede?
54. Numa sub-rede classe C sabemos que o endereço de um *host* e da máscara são:
- Endereço IP: 192.44.82.16
Máscara: 255.255.255.192
- Qual é o endereço dessa sub-rede?
55. Determine as máscaras para criar as seguintes quantidades de sub-redes classe A.

- a. 2
 - b. 6
 - c. 30
 - d. 62
 - e. 122
 - f. 250
56. Determine as máscaras para criar as seguintes quantidades de sub-redes classe B.
- a. 2
 - b. 5
 - c. 30
 - d. 62
 - e. 120
 - f. 250
57. Qual é o número máximo de sub-redes classe A que podem ser criadas a partir das seguintes máscaras?
- a. 255.255.192.0
 - b. 255.192.0.0
 - c. 255.255.224.0
 - d. 255.255.255.0
58. Qual é o número máximo de sub-redes classe B que podem ser criadas a partir das seguintes máscaras?
- a. 255.255.192.0
 - b. 255.255.0.0
 - c. 255.255.224.0
 - d. 255.255.255.0
59. Qual é o número máximo de sub-redes classe C que podem ser criadas a partir das seguintes máscaras?
- a. 255.255.255.192
 - b. 255.255.255.224
 - c. 255.255.255.240
 - d. 255.255.255.0
60. Para cada uma das seguintes máscaras de sub-redes usadas na classe A, determine a quantidade de 1s que definem a sub-rede.
- a. 255.255.192.0
 - b. 255.192.0.0
 - c. 255.255.224.0
 - d. 255.255.255.0
61. Para cada uma das seguintes máscaras de sub-redes usadas na classe B, determine a quantidade de 1s que definem a sub-rede.
- a. 255.255.192.0
 - b. 255.255.0.0
 - c. 255.255.224.0
 - d. 255.255.255.0
62. Para cada uma das seguintes máscaras de sub-redes usadas na classe C, determine a quantidade de 1s que definem a sub-rede.
- a. 255.255.255.192
 - b. 255.255.255.224
 - c. 255.255.255.240
 - d. 255.255.255.0
63. Escreva as seguintes máscaras de sub-redes no formato /*n*.
- a. 255.255.255.0
 - b. 255.0.0.0
 - c. 255.255.224.0
 - d. 255.255.240.0
64. Determine a faixa de endereços IP dados a um endereço de rede e a máscara no formato /*n*.
- a. 123.56.77.32/29
 - b. 200.17.21.128/27
 - c. 17.34.16.0/23
 - d. 180.34.64.64/30

Protocolos da Camada de Rede: ARP, IPv4, ICMP, IPv6 e ICMPv6

No modelo da Internet, ou arquitetura TCP/IP, há cinco protocolos que operam na camada de rede: ARP, RARP, IP, ICMP e IGMP, conforme ilustra a Figura 20.1.



Figura 20.1 Protocolos da camada de rede.

Sem dúvida o principal protocolo da camada de rede é o IP, responsável pelos processos de entrega *host-to-host* de datagramas da origem ao destino. Entretanto, o protocolo IP necessita dos serviços de outros protocolos.

O IP precisa de um protocolo denominado ARP para determinar o endereço físico (MAC) do próximo salto. Este endereço deve ser passado à camada de enlace, através do datagrama IP, para ser inserido no *frame* encapsulado.

Durante o processo de entrega do datagrama, o IP chama os serviços do protocolo ICMP para controlar situações extraordinárias, tal como a ocorrência de erros.

O IP foi criado para promover a entrega de pacotes *unicast*, isto é, entre uma origem e um destino. As aplicações de multimídia e outras novas aplicações da Internet necessitam de processos de entregas de pacotes *multicast*, envolvendo uma origem e muitos destinos. Para os processos de *multicasting*, o IP solicita os serviços de outro protocolo denominado IGMP.

Neste capítulo discutiremos somente o ARP, IP e o ICMP. O protocolo RARP hoje está obsoleto. O IGMP será discutido no Capítulo 21, quando estivermos tratando os processos *multicasting*.

A versão atual do IP é denominada IPv4. A nova versão, que pode ou não vir a predominar, é a IPv6. No final deste capítulo daremos uma olhadela neste novo protocolo e racionalizaremos sobre a existência dele.

20.1 ARP

A Internet é constituída de uma combinação de redes físicas conectadas por vários dispositivos, tal como roteadores. Um pacote saindo de um *host* de origem pode passar através de muitas redes físicas diferentes antes de alcançar o *host* de destino.

Os *hosts* (incluindo os roteadores) são identificados no nível da camada de rede através dos respectivos endereços IP. Um **endereço IP** é um endereço de *internetworking*. O IP possui jurisdição universal. Desse modo, um endereço IP válido é universalmente único. Todo protocolo de camada superior que cuida de algum processo nas *internetworks* conectadas à Internet precisam do endereço IP.

Contudo, os pacotes antes de alcançar *hosts* ou roteadores dentro de uma rede passam através das redes físicas. No nível das redes físicas, os *hosts* (incluindo os roteadores) são reconhecidos através dos respectivos endereços físicos (MAC). Um endereço MAC é um endereço local. A jurisdição dele é a rede local. Este endereço deve ser único apenas localmente, mas não é necessariamente universal.

Os endereços MAC e IP possuem dois tipos de identificadores diferentes. Precisamos dos dois protocolos porque uma rede física, como a Ethernet, geralmente utiliza dois ou mais protocolos diferentes de camada de rede ao mesmo tempo, tal como o IP e o IPX (Novell). Do mesmo modo, um pacote como o IP pode trafegar por diferentes redes físicas, por exemplo, Ethernet e Token Ring.

Isto significa que o processo de entrega de um pacote para um *host* ou roteador requer dois níveis de endereçamento: IP e MAC. Precisamos ser capazes de mapear um endereço IP a partir do endereço MAC correspondente.

Mapeamento

Existem dois tipos de mapeamento de endereços: estático e dinâmico.

Mapeamento Estático

O **mapeamento estático** é criado quando um administrador da rede entra manualmente na tabela de endereçamento de um *host* e associa os endereços IP aos endereços MAC correspondentes. Esta tabela fica armazenada em cada *host* da rede. Todo *host* que conhece, por exemplo, o endereço IP de outro *host*, mas desconhece o endereço MAC desse mesmo *host*, pode efetuar uma pesquisa na tabela de endereçamento para tentar consegui-lo. Isto tem algumas limitações porque os endereços MAC podem ser modificados de muitas formas:

1. O adaptador de rede do *host* pode ter sido substituído, devido a danos, resultando em um novo endereço MAC.
2. Em algumas LANs, tal como a LocalTalk (Apple), o endereço MAC muda toda vez que o computador é ligado.
3. Os computadores portáteis (*notebook* ou *laptop*) têm liberdade de se mover entre as diversas redes físicas, introduzindo novos endereços físicos nas redes de destino toda vez que são conectados a elas.

Para suportar todas essas variáveis, o endereçamento estático deve ser atualizado periodicamente. Tabelas desatualizadas afetam a *performance* de uma rede.

Mapeamento Dinâmico

No mapeamento dinâmico, toda vez que um *host* conhece um dos dois endereços, ele pode usar um protocolo dinâmico para determinar o outro.

Existem dois protocolos criados especialmente para realizar o mapeamento dinâmico: **Address Resolution Protocol (ARP)** e o **Reverse Address Resolution Protocol (RARP)**. O ARP mapeia um endereço IP no respectivo endereço MAC. O RARP faz o inverso, isto é, mapeia um endereço MAC a um endereço IP. Contudo, discutiremos somente o ARP porque o RARP está sendo substituído sistematicamente pelo DHCP (veja Capítulo 19).

Como dissemos, o ARP mapeia (associa) um endereço IP a um endereço MAC. Numa rede física típica, como uma LAN Ethernet, cada dispositivo da rede é identificado fisicamente através do endereço que vem gravado no adaptador de rede (NIC – Network Interface Card).

De qualquer forma, um *host* (incluindo a interface do roteador) que precisa determinar o endereço MAC de outro *host* na rede, envia um pacote ARP de consulta (*query*). Este pacote inclui o endereço físico e IP da origem e o endereço IP do destino. Como o *host* de origem não conhece o endereço físico do *host* de destino, ele transmite a consulta em modo *broadcast* através da rede, para alcançar todos os *hosts* da rede física ao mesmo tempo (veja a Figura 20.2).

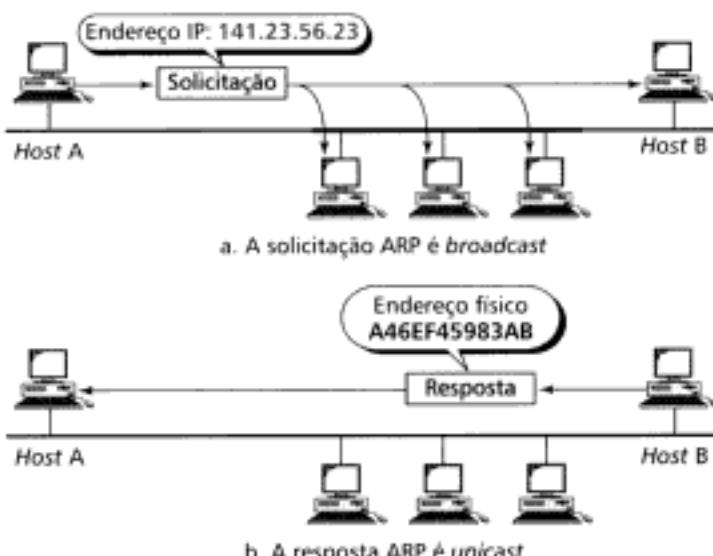


Figura 20.2 Operação ARP.

Como os pacotes de solicitação ARP trafegam em modo *broadcast*, todos os dispositivos na rede local recebem esses pacotes e os passam à camada da rede para que sejam examinados. Se o endereço IP de um dispositivo coincidir com o endereço IP de destino na consulta ARP, esse dispositivo responde, em modo *unicast*, enviando seu endereço MAC à origem. Isso é conhecido como resposta ARP.

Na Figura 20.2a, o *host* da esquerda (A) possui um pacote que precisa ser entregue ao outro *host* (B), cujo endereço IP é 141.23.56.23. O *host* A necessita enviar o pacote para a camada de enlace do dispositivo para o qual deseja transmitir, mas ele desconhece o endereço físico do receptor. Desse modo, ele usa os serviços do protocolo ARP para enviar um pacote de solicitação em modo *broadcast* perguntando sobre o endereço físico do *host* cujo endereço IP é 141.23.56.23.

Este pacote é recebido por todos os *hosts* do sistema físico, mas somente o *host* que tiver o endereço IP supracitado, isto é, o *host* B, irá responder (veja a Figura 20.2b). O *host* B envia uma resposta ARP que inclui, dentre outras informações, o endereço físico dele. Assim, o *host* A aprende o endereço físico de B e o registra na tabela ARP. Nesse caso, o *host* A pode transmitir todos os pacotes ao *host* B usando o endereço físico fornecido por B.

Formato do Pacote ARP

A Figura 20.3 mostra o formato de um pacote ARP típico.

Os campos do pacote ARP são os seguintes:

- **HTYPE (tipo de hardware).** Este campo tem 16-bits de tamanho definindo o tipo de rede por onde o ARP está trafegando. Para cada tipo de LAN foi atribuído um número inteiro de identificação. Por exemplo, HTYPE para a rede Ethernet é dado o tipo 1. O ARP pode ser utilizado em qualquer rede física.

Tipo de hardware		Tipo de protocolo
Tamanho do endereço físico	Tamanho do protocolo	Operação solicitação 1, resposta 2
Endereço físico de origem (por exemplo, 6 bytes para Ethernet)		
Protocolo do endereço de origem (por exemplo, 4 bytes para IP)		
Endereço físico do destino (por exemplo, 6 bytes para Ethernet) (não é preenchido numa solicitação)		
Protocolo do endereço de destino (por exemplo, 4 bytes para IP)		

Figura 20.3 Pacote ARP.

- **PTYPE (tipo de protocolo).** Este campo tem 16-bits de tamanho definindo o tipo de protocolo que está utilizando o ARP. Por exemplo, o valor deste campo para a versão IPv4 é 0800H. O ARP pode ser utilizado por qualquer protocolo de camada superior.
- **HLEN (tamanho do endereço físico).** Este campo tem 8-bits definindo o tamanho do endereço físico em bytes. Por exemplo, para a rede Ethernet o valor deste campo é 6.
- **PLEN (tamanho do protocolo).** Este campo tem 8-bits definindo o tamanho do endereço IP em bytes. Por exemplo, para a versão IPv4 o valor deste campo é 4.
- **OPER (operação).** Este campo tem 16-bits definindo o tipo de pacote. Dois tipos de pacotes estão definidos: solicitação ARP (1) e resposta ARP (2).
- **SHA (endereço físico de origem).** Este campo possui tamanho variável definindo o endereço físico do host de origem. Por exemplo, para a rede Ethernet este campo possui um tamanho 6 bytes.
- **SPA (protocolo do endereço de origem).** Este campo possui um tamanho variável definindo o endereço lógico (por exemplo o IP) do host de origem. Para o protocolo IP, este campo possui um tamanho de 4-bytes.
- **THA (endereço físico de destino).** Este campo possui um tamanho variável definindo o endereço físico do host alvo. Por exemplo, para a rede Ethernet este campo possui um tamanho de 6-bytes. Para uma solicitação ARP, este campo possui 0s em todas as posições porque o host de origem não sabe para quem enviar a mensagem.
- **TPA (protocolo do endereço de destino).** Este campo possui um tamanho variável definindo o endereço lógico (por exemplo o IP) do host alvo. Para a versão IPv4, este campo tem 4-bytes de tamanho.

Encapsulamento

Um pacote ARP é encapsulado diretamente em um frame na camada de enlace. Por exemplo, na Figura 20.4, um pacote ARP é encapsulado em um frame Ethernet. Perceba que o campo de tipo indica que os dados transportados no frame são de um pacote ARP.

Operação

Vamos ver como funciona o ARP na Internet. Inicialmente, descrevemos as etapas envolvidas. Então, descreveremos quatro situações diferentes onde um host (incluindo um roteador) precisa utilizar o ARP.

Hidden page

Hidden page

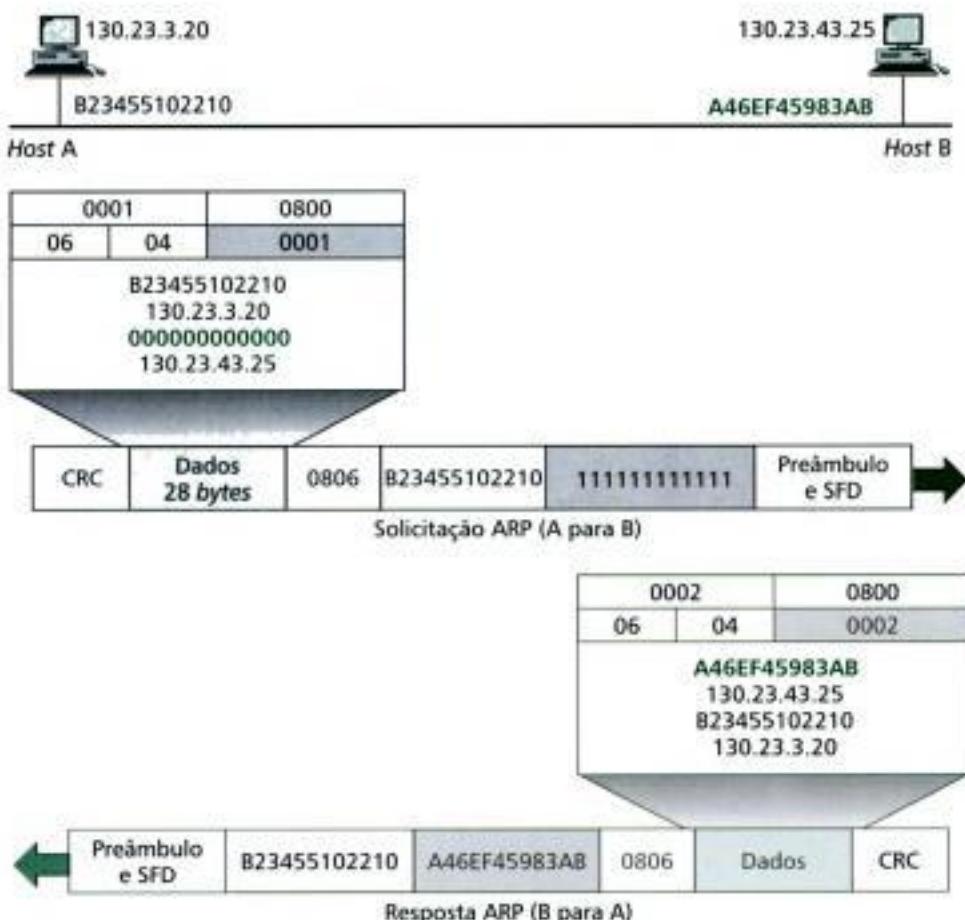


Figura 20.6 Exemplo 1.

não tem como saber que ele deveria ter recebido uma correspondência não registrada. Os Correios, na maioria das vezes, não mantêm um registro desse tipo de correspondência e não conseguem notificar o remetente da perda, dano ou extravio da correspondência.

O IP também é protocolo sem conexão para redes de comutação de pacotes que utilizam a abordagem de datagramas (veja Capítulo 19). Isto significa que cada datagrama é controlado independentemente e pode seguir por uma rota diferente até o destino. Assim, os datagramas que saem de uma origem podem chegar fora de ordem no destino. Além disso, alguns dos pacotes podem ser perdidos durante a transmissão. Novamente, o IP confia no protocolo de camada superior (camada de transporte) para resolver todos estes problemas.

Datagrama

Os pacotes na camada IP são denominados **datagramas**. A Figura 20.7 mostra o formato do datagrama IP. Um datagrama IP é um pacote de tamanho variável consistindo de duas partes: cabeçalho e dados. O **cabeçalho** possui um tamanho que varia de 20 a 60-bytes e contém informação essencial para os serviços de roteamento e entrega. É praxe no estudo desse protocolo mostrar o cabeçalho em seções de 4-bytes. Uma breve descrição de cada campo é feita na ordem em que aparecem no datagrama.

- **VER (versão).** Este campo define a versão do protocolo IP utilizado. Atualmente, a versão é a 4 (IPv4). Entretanto, espera-se que a versão 6 (IPv6) substitua completamente a versão 4 num futuro muito próximo.
- **HLEN (tamanho do cabeçalho).** Graças a este campo o tamanho do cabeçalho é variável. Este campo define o tamanho do cabeçalho do datagrama em palavras de 4-bytes. O valor do campo deve ser multiplicado por 4 para se obter o tamanho real em bytes.

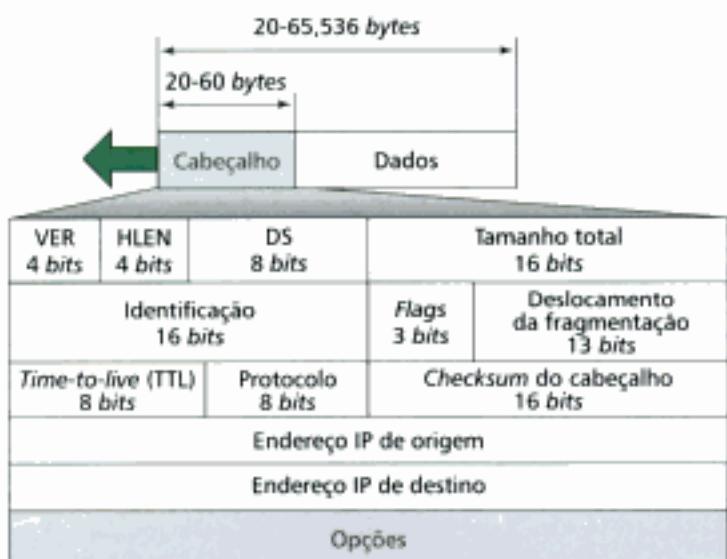


Figura 20.7 Datagrama IP.

- **DS (serviços diferenciados).** Este campo define a classe de datagrama para os propósitos da qualidade de serviços (QoS). Discutimos a QoS no Capítulo 23.
- **Total length (tamanho total).** Este campo define o tamanho total do datagrama IP (cabeçalho + dados). Para determinar o tamanho do campo de dados que chegam da camada superior, subtraia o tamanho do cabeçalho do tamanho total. Como dissemos, o tamanho do cabeçalho pode ser determinado multiplicando o valor do campo HLEN por 4.

Tamanho do campo de dados = tamanho total – tamanho do cabeçalho

Como o campo tamanho tem 16 bits, o tamanho total do datagrama IP fica limitado a 65.535 ($2^{16} - 1$) bytes, dos quais de 20 a 60 bytes formam o cabeçalho e o resto é reservado para dados da camada superior (*payload*).

O campo tamanho total do datagrama especifica o tamanho total do pacote IP, incluindo dados e cabeçalho.

Embora 65.535 bytes de tamanho possa parecer muito grande, o tamanho do datagrama IP ainda deve aumentar num futuro próximo, quando as tecnologias subjacentes permitirem um throughput maior (usando larguras de bandas maiores).

- **Identificação, flag e offset.** Na próxima seção discutiremos estes três campos quando tratarmos a fragmentação.
- **TTL (time-to-live).** Este campo é usado para controlar o número máximo de saltos (roteadores) visitados pelo datagrama. Quando um host de origem envia um datagrama um número é armazenado neste campo. O valor é aproximadamente 2 vezes o número máximo de rotas entre os dois hosts. Cada roteador que processar o datagrama decrementa este número de 1 unidade. Se este valor, após sofrer consecutivos decréscimos, chegar a zero, o roteador que o estiver processando descarta-o. O propósito desse campo é evitar que o datagrama torne-se um pacote errante, vagando de um roteador para outro sem chegar ao destino.
- **Protocolo.** Este campo define o protocolo de camada superior que usa os serviços da camada IP. Um datagrama IP pode encapsular dados de diversos protocolos, tal como o TCP, UDP, ICMP e IGMP. Este campo indica que protocolo de camada superior receberá os pacotes de entrada depois que o processamento do IP tiver sido concluído. Em outras palavras, como o IP multiplexa e demultiplexa dados oriundos de diferentes protocolos de camada superior, o valor deste campo ajuda no processo de demultiplexação quando o datagrama chegar ao destino final (veja Figura 20.8).

Hidden page

- **Endereço de destino.** Este campo define o endereço IP do *host* de destino. O valor desse campo deve permanecer constante durante o tempo de viagem do datagrama IP do *host* de origem ao *host* de destino.
- **Opções.** As opções, como sugere o nome, não são campos obrigatórios para os datagramas. Elas são utilizadas para teste e depuração (*debugging*) da rede. Embora as opções não sejam uma parte requerida do cabeçalho IP, o processamento dessas opções é realizado pelo programa que verifica o cabeçalho IP. Há muitos tipos de opções que podem ser controladas nesse campo, mas não as discutiremos aqui. Para mais informações, veja Forouzan, *TCP/IP Protocols Suite*, 2d ed., McGraw-Hill, 2002.

Fragmentação

Já sabemos que um datagrama IP pode viajar através de muitos tipos de redes diferentes. Um roteador desencapsula um datagrama do *frame* recebido e volta a encapsulá-lo novamente noutro *frame*. O formato e o tamanho do *frame* recebido depende do protocolo usado pela rede física através da qual o datagrama viaja. Da mesma forma, o formato e o tamanho do *frame* transmitido depende do protocolo usado pela rede física através da qual o datagrama irá passar. Por exemplo, se um roteador conecta uma rede Ethernet a uma rede ATM, o *frame* é recebido no formato Ethernet e é transmitido no formato ATM.

Unidade Máxima de Transmissão (MTU)

Os protocolos da camada de enlace possuem um formato próprio. Um dos campos definidos no formato é o tamanho máximo do campo de dados. Assim, quando um datagrama é encapsulado em um *frame*, o tamanho total do datagrama deve ser menor que o tamanho máximo previsto na camada de enlace, que é definido por restrições impostas pelo *hardware* e *software* usado na rede (veja a Figura 20.10).



Figura 20.10 MTU.

Para tornar o IP independente da rede física, os especialistas em pacotes decidiram tornar o tamanho máximo do datagrama IP igual ao maior valor da MTU (Maximum Transfer Unit) definida até o momento (65.535 bytes). Isto torna uma transmissão mais eficiente se usarmos um protocolo com este tamanho de MTU. Contudo, para outras redes físicas, devemos dividir o datagrama para possibilitar a passagem dele através delas. Este processo de divisão é denominado **fragmentação**.

Quando um datagrama é fragmentado, todo fragmento carrega um cabeçalho próprio com a maioria dos campos repetidos e que são encontrados em outros fragmentos, mas alguns campos são modificados. Até mesmo um datagrama fragmentado pode sofrer nova fragmentação, se ele encontrar uma rede física com um valor menor de MTU. Sendo assim, um datagrama pode ser fragmentado tantas vezes quanto se fizerem necessárias até que ele atinja o destino final.

Um datagrama pode ser fragmentado pelo *host* de origem ou por qualquer roteador ao longo do caminho. Entretanto, o reagrupamento do datagrama é feito somente no destino final, pois cada datagrama é uma entidade independente, não tendo que passar todos necessariamente pelos mesmos caminhos. Considerando que o datagrama fragmentado possa passar por diferentes rotas e que não temos como controlar ou garantir a rota que um datagrama fragmentado deve passar,

todos os fragmentos pertencentes ao mesmo datagrama devem chegar ao *host* de destino para serem reagrupados. Então, parece óbvio que o único local onde é possível fazer o reagrupamento do datagrama é no destino final.

Campos Envolvidos na Fragmentação

Os campos envolvidos no processo de fragmentação e reagrupamento do datagrama IP são os campos de identificação, *flags* e de deslocamento de fragmento.

- **Identificação.** Este campo identifica um datagrama originado em um *host* de origem. Quando um datagrama é fragmentado, o valor indicado no campo de identificação é copiado em todos os fragmentos. Sendo assim, todos os fragmentos carregam o mesmo número de identificação, que também são idênticos ao número de identificação do datagrama original. O número de identificação auxilia o *host* de destino no processo de reagrupamento do datagrama. O *host* de destino sabe que todos os fragmentos sob a tutela do mesmo número de identificação pertencem a um único datagrama.
- **Flags.** Este é um campo de 3-bits. O primeiro bit é reservado. O segundo bit é denominado *não fragmentar*. Se o valor desse campo é 1, o *host* não deve fragmentar o datagrama. Se um *host* (roteador) no meio do caminho não puder passar o datagrama pela rede física de interesse sem fragmentação, o *host* irá descartar o datagrama e enviar uma mensagem de erro ICMP ao *host* de origem (próxima seção). Se o valor desse campo é 0, o datagrama pode ser fragmentado, se necessário. O terceiro bit é denominado *mais fragmentos*. Se o valor deste bit é 1, significa que o datagrama não é o último fragmento, ou seja, o *host* pode receber outros fragmentos depois desse. Se o valor do campo é 0, significa que este é o último ou o único fragmento.
- **Deslocamento do fragmento.** Este é um campo de 13-bits que mostra a posição relativa de um fragmento com relação ao datagrama como um todo. Ele representa o deslocamento dos dados, em unidades de 8-bytes, em relação à medida do datagrama original. A Figura 20.11 mostra um datagrama com uma área de dados de 4000 bytes fragmentada em três partes. Os bytes do datagrama original são numerados de 0 a 3999. O primeiro fragmento transporta os bytes de 0 a 1399. O deslocamento para este datagrama é $0/8 = 0$. O segundo fragmento transporta os bytes de 1400 a 2799. O valor do deslocamento para este fragmento é $1400/8 = 175$. Finalmente, o terceiro fragmento transporta os bytes de 2800 a 3999. O valor do deslocamento para este fragmento é $2800/8 = 350$.

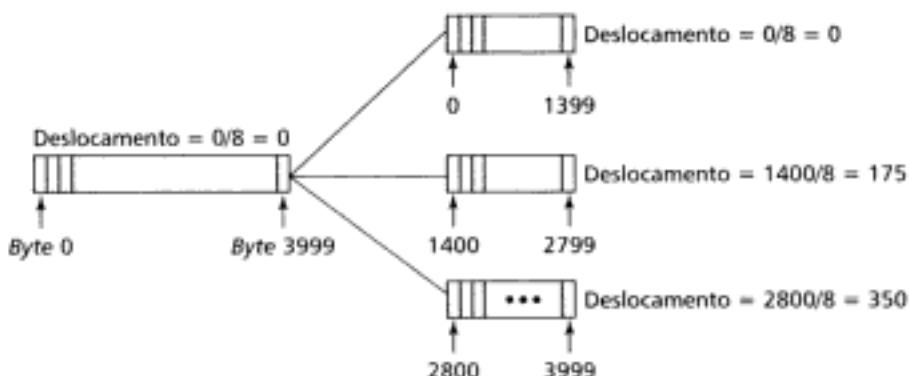


Figura 20.11 Exemplo de fragmentação.

Lembre-se que o valor do deslocamento é medido em unidade de 8-bytes. Isto é assim porque o tamanho do campo de deslocamento é somente 13-bits e não pode representar uma sequência de bytes maior que 8191. Isto força hosts (incluindo roteadores) que fragmentam datagramas a escolherem o tamanho de cada fragmento de forma que o primeiro número de bytes seja divisível por 8.

Hidden page

O ICMP sempre reporta mensagens de erros ao *host* de origem.

Cinco tipos de erros são controlados pelo ICMP: destino inalcançável, tráfego, tempo excedido, problema nos parâmetros e redirecionamento (veja a Figura 20.13).



Figura 20.13 Mensagens reportando erros.

Mensagem de Destino Inalcançável Quando um roteador não pode rotear um datagrama ou um *host* não pode entregar um datagrama, o datagrama é descartado e o roteador ou *host* envia uma **mensagem de destino inalcançável (*unreachable destination*)** de volta ao *host* de origem.

Mensagem de Tráfego O IP é um protocolo sem conexão. Não há nenhuma comunicação antecipada entre o *host* de origem, que produz o datagrama, os roteadores que roteiam o datagrama e o *host* de destino que processa o datagrama. Uma das implicações dessa ausência de comunicação é a falta de *controle de fluxo* e de *congestionamento* (veja Capítulo 23). A ausência de controle de fluxo pode criar um problema ainda maior no processo de comunicação origem-destino. O *host* de origem, sem saber que o *host* de destino está congestionado, continua transmitindo datagramas. A falta de controle de congestionamento também pode criar um problema sério nos roteadores que estão supostamente no caminho dos pacotes.

Não existem mecanismos de controle de fluxo e de congestionamento no protocolo IP.

A **mensagem de tráfego (*source-quench*)** do ICMP foi criada para agregar um tipo de controle de fluxo e de congestionamento ao IP. Quando um roteador ou *host* descarta um datagrama, pelo excesso de congestionamento, uma mensagem de tráfego é transmitida ao *host* de origem do datagrama. Esta mensagem tem dois propósitos. Primeiro, informar ao *host* de origem que o datagrama foi descartado. Segundo, advertir o *host* de origem sobre o congestionamento em algum lugar no caminho e solicitar ao *host* de origem que diminua a taxa de transmissão de pacotes para, possivelmente, diminuir congestionamento na rede.

Mensagem de Tempo Excedido A mensagem de tempo excedido (*time-exceeded*) é gerada em dois casos. Primeiro, o roteador que recebe um datagrama com o campo TTL com um valor 0 descarta o datagrama. Porém, no ato do descarte do datagrama uma mensagem de tempo excedido é transmitida do roteador ao *host* de origem. Segundo, uma mensagem de tempo excedido também é gerada quando todos os fragmentos que compõem uma mensagem não chegam ao *host* de destino dentro de um tempo limite.

Mensagem de Problema nos Parâmetros Qualquer ambigüidade na parte do cabeçalho de um datagrama pode criar sérios problemas quando o datagrama viaja através da Internet. Se um roteador ou *host* de destino descobre uma ambigüidade qualquer ou um valor perdido no campo de cabeçalho do datagrama, eles descartam o datagrama e enviam uma **mensagem de problema nos parâmetros (*parameter-problem*)** de volta ao *host* de origem.

Mensagem de Redirecionamento Quando um roteador precisa enviar um pacote destinado a outra rede, ele deve saber, antecipadamente, o endereço IP do próximo salto (roteador). O mesmo é verdadeiro se o transmissor for um *host*. Ambos, roteadores e *hosts*, devem possuir uma tabela de

roteamento para determinar o endereço do próximo salto. Os roteadores tomam parte no processo de atualização do roteamento, como veremos no Capítulo 21, e supostamente estão sempre atualizados, pois o roteamento é dinâmico.

Contudo, para melhorar a eficiência da rede, os *hosts* não tomam parte na processo de atualização porque há muito mais *hosts* (computadores) numa rede do que roteadores. A atualização dinâmica da tabela de roteamento dos *hosts* produziria uma tráfego inaceitável. Os *hosts* usualmente usam roteamento estático. Quando um *host* é colocado pela primeira vez na rede, a tabela de roteamento dele possui um número limitado de entradas. Geralmente, ele conhece somente o endereço IP de um roteador (o *gateway* padrão da rede). Por esta razão, o *host* pode enviar um datagrama, destinado a outra rede, para o roteador errado. Nesse caso, o roteador que receber o datagrama redirecionará o datagrama ao roteador correto. Entretanto, para que a tabela de roteamento do *host* seja atualizada, o roteador que redirecionou o datagrama envia uma **mensagem de redirecionamento (redirection)** de volta ao *host*.

Query

Além de reportar os erros, o ICMP pode diagnosticar alguns problemas da rede. Isto é realizado através das **mensagens de query (consulta)**, um grupo de quatro pares de mensagens diferentes, conforme ilustra a Figura 20.14. Neste tipo de mensagem ICMP, um nó origem envia uma mensagem que é respondida num formato específico pelo nó destino.



Figura 20.14 Mensagens de consulta (query).

Mensagem de Solicitação e de Resposta de Eco As mensagens de **solicitação de eco (echo-request)** e **resposta de eco (echo-reply)** foram criadas para os propósitos de diagnóstico. Os administradores de redes e alguns usuários utilizam estas mensagens para identificar problemas na rede. A combinação das mensagens de solicitação de eco e resposta de eco determina se dois sistemas (*hosts* ou roteadores) podem comunicar-se.

Mensagem de Solicitação e de Resposta Time-Stamp Dois *hosts* (incluindo roteadores) podem usar as mensagens de **solicitação time-stamp** e de **resposta time-stamp** para determinar o tempo necessário para um datagrama IP viajar entre eles. Estas mensagens também podem ser utilizadas para sincronizar os relógios dos dois *hosts*.

Mensagem de Solicitação e de Resposta de Endereço de Máscara O endereço IP de um *host* contém um endereço da rede, endereço da sub-rede e identificador de *host*. Um *host* pode conhecer totalmente o endereço IP dele, mas pode não saber que parte do endereço define os endereços da rede e da sub-rede e que parte corresponde ao identificador do *host*. Neste caso, o *host* pode enviar uma mensagem de **solicitação do endereço de máscara (address mask request)** a um roteador. O roteador envia então uma máscara na mensagem de **resposta do endereço de máscara (address mask reply)**.

Mensagem de Solicitação e de Anúncio do Roteador Conforme discutimos na seção sobre a mensagem de redirecionamento, um *host* que tiver dados a transmitir para outro *host* localizado fora da rede onde reside precisa conhecer o endereço dos roteadores conectados nessas duas redes. Além disso, o *host* deve saber se os roteadores estão ativos e em funcionamento. A mensagem de **solicitação do roteador (router-solicitation)** e a mensagem de **anúncio do roteador (router-advertisement)** podem auxiliar nestas questões. Um *host* pode enviar uma mensagem de broadcast (*multicast*) de solicitação do roteador. O roteador ou roteadores que receberem a mensagem transmi-

te(m) a informação de roteamento de que dispõem através da mensagem de anúncio do roteador. Um roteador também pode enviar mensagens de anúncio periodicamente até mesmo se não tiver sido solicitada. Devemos notar que, quando um roteador anuncia a presença dele, na verdade está anunciando a presença de todos os roteadores que ele tem conhecimento na rede.

20.4 IPv6

Atualmente, o protocolo de camada de rede na Internet é o **IPv4**. O IPv4 proporciona comunicação *host-to-host* entre sistemas na Internet. Embora o IPv4 tenha sido bem projetado, a comunicação de dados evoluiu desde o início da década de 70, quando o IPv4 foi criado. O IPv4 tem algumas deficiências que o tornam inconveniente para a Internet como ela é hoje. Dentre as deficiências podemos citar:

- IPv4 possui dois níveis de estrutura de endereços (*netid* e *hostid*) distribuídos em cinco classes (A,B,C,D e E). O uso da faixa de endereços é ineficiente.
- A Internet precisa fornecer transmissão de áudio e vídeo em tempo real. Este tipo de transmissão requer estratégias de atraso mínimas e reserva de recursos não encontrados no IPv4.
- A Internet precisa incorporar criptografia e processos de autenticação de dados para algumas aplicações. Originalmente, nenhum mecanismo de segurança foi provido pelo IPv4.

Para sobrepujar tais deficiências foi criado, e já encontra-se devidamente padronizado, o **Internet Protocol versão 6 (IPv6)**, também conhecido como **Internetworking Protocol, next generation (IPng)**. No IPv6, o protocolo de Internet foi extensivamente modificado para incorporar o inesperado crescimento da Internet. O formato e o tamanho dos endereços IP foram modificados, tanto quanto o formato do pacote.

O IPv6 possui inherentemente algumas vantagens sobre o IPv4 que sintetizaremos abaixo:

- **Maior faixa de endereçamento.** Um endereço IPv6 possui um tamanho de 128-bits. Compare com os 32-bits de endereçamento do IPv4. O aumento na quantidade de endereços é da ordem de 2^{96} .
- **Melhor formato do cabeçalho.** O IPv6 usa um novo formato de cabeçalho onde as opções foram separadas do cabeçalho base e inseridas, quando necessárias, entre o cabeçalho base e a camada superior. Isto simplifica e aumenta, incrivelmente, a velocidade dos processos de roteamento porque a maioria das opções não precisam ser verificadas pelos roteadores.
- **Novas opções.** IPv6 agrupa novas opções que permitem adicionar outras funcionalidades ao protocolo.
- **Tolerância à extensão.** O IPv6 foi projetado de modo a permitir a extensão do protocolo, caso novas tecnologias ou aplicações venham requerer.
- **Supporte à alocação de recursos.** No IPv6 o campo tipo de serviço foi removido, mas um mecanismo denominado **flow label** foi adicionado para habilitar o *host* de origem a exercer controle sobre pacotes especiais. Este mecanismo consegue suportar tráfego como áudio e vídeo em tempo real.
- **Supporte à segurança.** As opções de criptografia e autenticação no IPv6 proporcionam confidencialidade e integridade aos pacotes.

Endereços IPv6

Um endereço IPv6 consiste de 16-bytes (octetos), isto é, ele tem 128-bits de tamanho (veja a Figura 20.15).

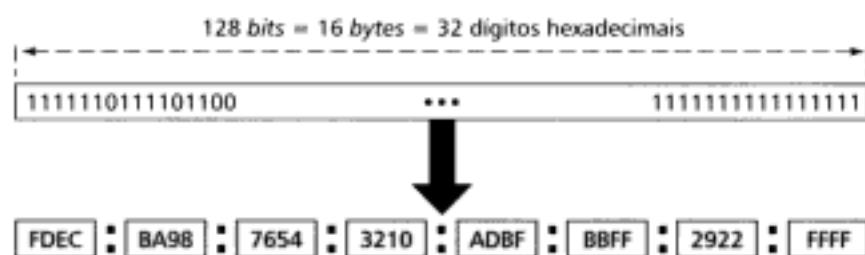


Figura 20.15 Endereço IPv6.

Notação Hexadecimal com dois Pontos

Para tornar os endereços um pouco mais inteligíveis, o IPv6 especifica a **notação hexadecimal com dois pontos**. Nesta notação, os 128-bits são divididos em oito seções, cada qual com dois bytes de tamanho. Dois bytes requerem quatro dígitos na notação hexadecimal. Assim, o endereço consiste de 32 dígitos hexadecimais, sendo cada 4 dígitos agrupados e separados por dois pontos.

Contração

Embora os endereços IP, até mesmo no formato hexadecimal, sejam extensos, muitos dos dígitos hexadecimais nos campos são zeros. Nestes casos, podemos **abreviar ou contrair os endereços**. Os zeros de uma seção (quatro dígitos entre dois pontos vizinhos) podem ser omitidos sempre que estiverem à esquerda dos números. Somente este tipo de zero pode ser omitido. Por exemplo, veja a Figura 20.16.

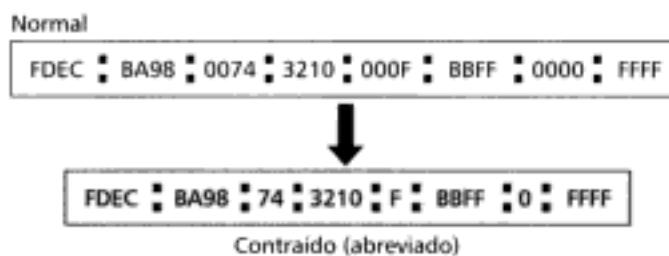


Figura 20.16 Endereço contraído (abreviado).

Usando este tipo de contração, 0074 pode ser escrito como 74, 000F como F e 0000 como 0. Perceba que o 3210 não pode ser abreviado. Outros tipos de contração são possíveis se houver seções consecutivas consistindo apenas de zeros. Podemos remover o conjunto de zeros e substituí-los por dois pontos emparelhados. A Figura 20.17 ilustra o conceito.

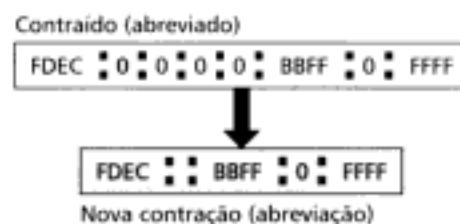


Figura 20.17 Endereços abreviado com zeros consecutivos.

Note que este tipo de contração é permitida somente uma vez por endereço. Se houver duas seções consecutivas em zero, somente uma delas poderá ser contraída. A expansão dos endereços contraídos é muito simples: alinhe as porções não abreviadas e insira zeros para obter o endereço original expandido.

Notação CIDR

O IPv6 permite endereçamento sem classes e a notação CIDR. Por exemplo, a Figura 20.18 mostra como podemos definir um prefixo de 60-bits usando o CIDR.

FDEC : 0 : 0 : 0 : 0 : BBFF : 0 : FFFF/60

Figura 20.18 Endereço CIDR.

Tipos de Endereços

O IPv6 define três tipos de endereços: *unicast*, *anycast* e *multicast*.

Endereços Unicast

Um endereço *unicast* define um único computador. O pacote enviado para um endereço *unicast* deve ser entregue ao computador específico.

Endereços Anycast

Um endereço *anycast* define um grupo de computadores com endereços tendo o mesmo prefixo. Por exemplo, todos os computadores conectados à mesma rede física compartilham o mesmo endereço de prefixo. Um pacote enviado para um endereço *anycast* deve ser entregue para exatamente um dos membros do grupo – o que estiver mais próximo ou mais facilmente acessível.

Endereços Multicast

Um endereço *multicast* define um grupo de computadores que podem ou não compartilhar o mesmo prefixo, e também, podem ou não estar conectados à mesma rede física. Um pacote enviado para um endereço *multicast* deve ser entregue a cada membro do grupo.

Formato do Pacote IPv6

O pacote IPv6 é mostrado na Figura 20.19. Cada pacote é composto de um cabeçalho base obrigatório seguido do *payload*. O *payload* consiste de duas partes: extensão do cabeçalho adicional e dados da camada superior. O cabeçalho base ocupa 40-bytes, considerando a extensão do cabeçalho e os dados da camada superior, o pacote tem 65.535 bytes de informação.

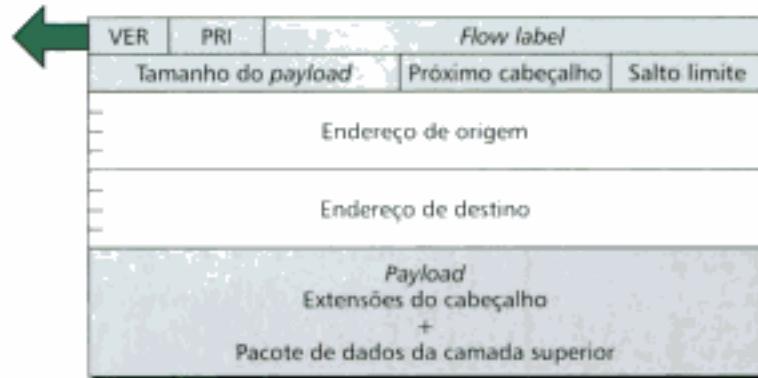


Figura 20.19 Formato de datagrama IPv6.

Cabeçalho Base

A Figura 20.19 ilustra o cabeçalho base com oito campos.

Estes campos são os seguintes:

- **Versão.** Este campo de 4-bits define o número da versão do IP. Para IPv6, o valor é 6.

- **Prioridade.** O campo de prioridade tem 4-bits e define o nível de prioridade do pacote com relação ao controle de congestionamento.
- **Flow label.** Este campo possui 3-bytes (24-bits) e foi desenvolvido para proporcionar controles especiais a fluxos particulares de informação, como áudio e vídeo em tempo real.
- **Tamanho do payload.** Este campo possui 2-bytes definindo o tamanho total do datagrama IP, excluindo o cabeçalho base.
- **Próximo cabeçalho.** Este campo possui 1-byte definindo o cabeçalho base no datagrama. Este campo contém uma extensão de cabeçalho usadas pelo IP ou pelo protocolo de camada superior, como o UDP e o TCP. Cada extensão de cabeçalho também possui este campo.
- **Salto limite.** Este campo de 8-bits serve aos mesmos propósitos que o campo TTL (*time-to-live*) na versão IPv4.
- **Endereço de origem.** Este campo possui um endereço de Internet de 16-bytes (128-bits) que identifica o *host* de origem do datagrama.
- **Endereço de destino.** Este campo possui um endereço de Internet de 16-bytes (128-bits) que usualmente identifica o destino final do datagrama. Entretanto, se o esquema de roteamento da origem for utilizado, este campo irá conter o endereço do próximo salto (roteador).

Extensão do Cabeçalho

O tamanho do cabeçalho base é fixo em 40-bytes. Contudo, para agregar mais funcionalidades ao datagrama IP, o cabeçalho base pode ser seguido por até seis **extensões do cabeçalho**. Muitos destes cabeçalhos são opções na versão IPv4. Mais detalhes podem ser encontrados em Forouzan, *TCP/IP Protocol Suite*, 2d ed., McGraw-Hill.

Fragmentação

O conceito de fragmentação é o mesmo utilizado na versão IPv4. Entretanto, o local onde a fragmentação acontece é diferente. No IPv4, um *host* de origem (incluindo um roteador) é requerido no processo de fragmentação do tamanho do datagrama, caso o tamanho do datagrama supere o MTU da rede onde ele irá trafegar. No IPv6, somente um *host* de origem de dados pode fragmentar datagramas. Antes de iniciar a comunicação, o *host* de origem deve utilizar uma técnica de descoberta do MTU do caminho para determinar o menor valor de MTU suportado pelas redes ao longo do caminho. O *host* de origem então fragmenta o datagrama baseado nessa informação.

Se o *host* de origem não fizer uso dessa técnica de descoberta de MTU, ele deve fragmentar o tamanho do datagrama em 576 bytes ou menos. Este é o tamanho do MTU mínimo requerido por cada rede conectada à Internet. A fragmentação no esquema IPv6 é controlada por uma das opções na extensão do cabeçalho.

Autenticação e Privacidade

O IPv6 suporta política de segurança, baseada na autenticação e privacidade, usando as opções na extensão do cabeçalho. Analisaremos as questões relativas à segurança de rede no Capítulo 31.

ICMPv6

Outro protocolo que sofreu uma modificação para a versão 6 da Internet foi o ICMP (**ICMPv6**). Esta nova versão segue as mesmas estratégias e propósitos da versão 4. O ICMPv4 foi modificado para incorporar novas funções mais apropriadas ao IPv6. Além disso, alguns protocolos que eram in-

dependentes na versão 4 fazem parte agora do ICMPv6. A Figura 20.20 compara as camadas de rede das versões 4 e 6 do modelo da Internet.

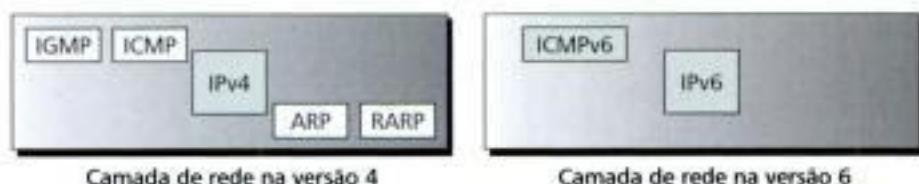


Figura 20.20 Comparação das camadas de rede das versões 4 e 6.

Os protocolos ARP e IGMP da versão 4 foram combinados ao ICMPv6. O RARP foi descartado pois raramente é utilizado nas aplicações atuais.

Migração do IPv4 para o IPv6

Como a quantidade de sistemas interligados à Internet é muito grande, a transição do IPv4 para o IPv6 não pode acontecer abruptamente. Certamente vai levar uma quantidade considerável de tempo antes que cada sistema da Internet possa migrar do IPv4 para o IPv6. A transição deve ser suave para prevenir quaisquer problemas entre o IPv4 e o IPv6.

O IETF planejou três estratégias de migração de modo a encurtar ou suavizar o tempo de migração (veja Figura 20.21).



Figura 20.21 Três estratégias de migração.

Dual Stack (Pilha dupla)

É bastante recomendável que todos os *hosts*, antes de migrar completamente para a versão 6, possam lidar com a pilha formada pelos dois protocolos simultaneamente, formando uma **dual stack**. Dessa maneira, cada estação deve rodar simultaneamente IPv4 e IPv6 até que toda a Internet tenha migrado para o IPv6. Observe que a Figura 20.22 mostra o *layout* da configuração *dual-stack*.

Para determinar que versão usar quando um *host* tiver que enviar um pacote ao destino, o *host* de origem faz uma consulta (*query*) de DNS (veja o Capítulo 25). Se o DNS retornar um endereço IPv4, o *host* de origem transmite um pacote IPv4. Se o DNS retornar um endereço IPv6, o *host* de origem transmite um pacote IPv6.

Tunelamento

Tunelamento é uma estratégia usada quando dois computadores usando IPv6 querem estabelecer comunicação entre eles através de um caminho que utiliza o IPv4. Para passar através desta região, o pacote deve possuir um endereço IPv4. Assim, o pacote IPv6 é encapsulado numa pacote IPv4 quando entrar na região e volta a ser desencapsulado em IPv6 quando sair dela. É como se o pacote IPv6 entrasse numa extremidade do túnel e emergisse na outra. Para sinalizar que o IPv4 está transportando um pacote IPv6 no campo de dados, o valor do campo protocolo é colocado em 41 (veja Figura 20.23).

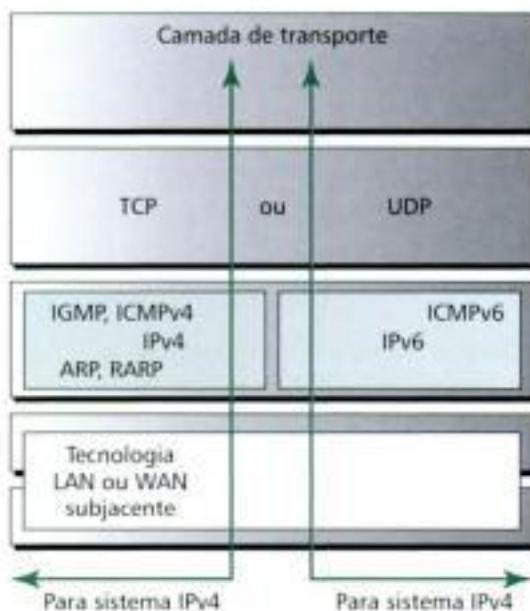


Figura 20.22 Dual stack.

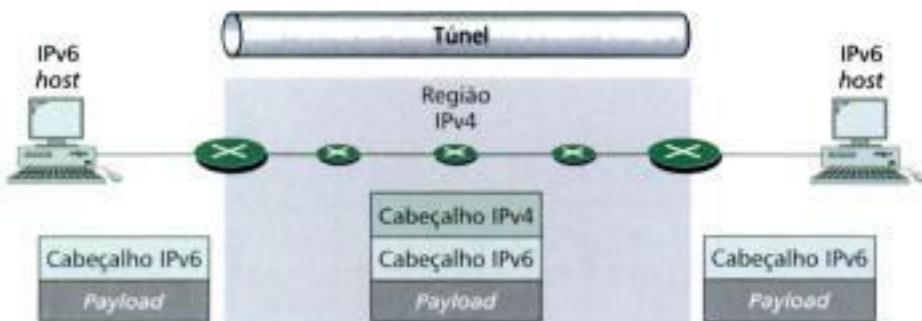


Figura 20.23 Tunelamento.

Tradução do Cabeçalho

A **tradução do cabeçalho** será necessária quando a maior parte da Internet tiver migrado para o IPv6, mas alguns poucos sistemas ainda usarem o IPv4. O processo será o seguinte: um *host* de origem deseja utilizar o IPv6 para se comunicar, mas o *host* de destino não pode compreendê-lo. A estratégia de tunelamento não funciona nessa situação porque o pacote deve estar no formato IPv4 para ser compreensível pelo receptor. Nesse caso, o formato do cabeçalho deve ser modificado totalmente através de uma tradução. O cabeçalho do pacote IPv6 é convertido num cabeçalho IPv4 (veja Figura 20.24).

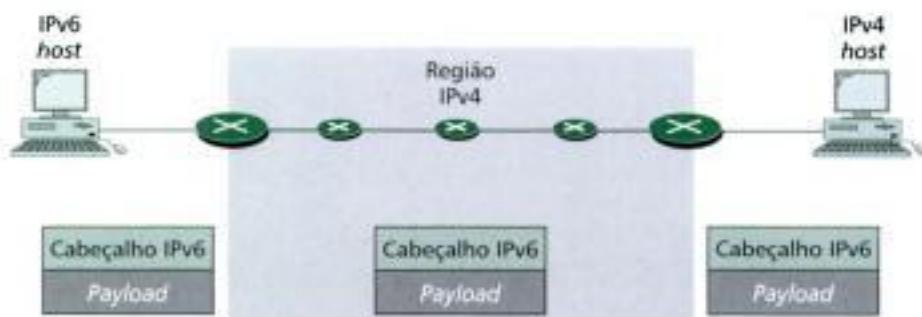


Figura 20.24 Tradução do cabeçalho.

Hidden page

Hidden page

Hidden page

Hidden page

- a. 2340:1ABC:119A:A000:0000:
0000:0000:0000
 - b. 0000:00AA:0000:0000:0000:
0000:119A:A231
 - c. 2340:0000:0000:0000:0000:
119A:A001:0000
 - d. 0000:0000:0000:2340:0000:
0000:0000:0000
58. Reescreva os seguintes endereços IPv6 no formato original.
- a. 0::0
 - b. 0:AA::0
 - c. 0:1234::3
 - d. 123::1:2
59. Quantos endereços a versão IPv6 tem a mais que a versão IPv4?

Hidden page

Roteamento Unicast e Multicast: Protocolos de Roteamento

Uma internet é a combinação de muitas redes conectadas através de roteadores. Quando um pacote é enviado para um destino específico, esse pacote passará provavelmente através de muitos roteadores até alcançar a rede de destino final. Um roteador consulta uma tabela de roteamento quando um pacote deve ser encaminhado através das portas dele. Uma tabela de roteamento especifica o melhor caminho para o pacote chegar a um destino. Elas podem ser de roteamento estático ou dinâmico. Uma *tabela de roteamento estático* não recebe, com frequência, muitas atualizações. Por outro lado, uma *de roteamento dinâmico* é atualizada automaticamente quando uma internet sofre modificações em algum lugar da rede. Atualmente, uma internet necessita de tabelas de roteamento dinâmicas para cuidar dos processos complexos de roteamento. Essas tabelas precisam ser atualizadas tão logo as mudanças na internet sejam detectadas. Por exemplo, as tabelas devem ser atualizadas sempre que um roteador ou *link* sofre um processo de *shutdown* (desligamento), físico ou lógico, ou então quando é criada uma rota melhor para um determinado destino.

Os protocolos de roteamento foram desenvolvidos em resposta às demandas por tabelas de roteamento dinâmicas. Um protocolo de roteamento combina regras e procedimentos que permitem aos roteadores de uma *internetworking* trocarem informações sobre o *status* e as mudanças de rotas na rede. Elas possibilitam também que os roteadores compartilhem o conhecimento que cada um tem sobre uma internet específica, a qual pertencem, e/ou sobre as rotas localizadas nas redes vizinhas. O compartilhamento da informação sobre as tabelas de roteamento permite que um roteador em San Francisco tome conhecimento da queda de uma rede no Texas. Os protocolos de roteamento também incluem procedimentos para combinar a informação recebida de outros roteadores.

Neste capítulo examinaremos dois tipos de roteamento: *unicast* e *multicast*. O roteamento *unicast* acontece entre um *host* de origem e um *host* destino, já o roteamento *multicast* envolve um *host* de origem e muitos *hosts* destinos.

21.1 ROTEAMENTO UNICAST

A comunicação *unicast* envolve apenas um *host* de origem e um *host* de destino. O relacionamento entre a origem e o destino é unívoco, isto é, um para um. Neste tipo de comunicação tanto o endereço IP de origem quanto o endereço IP de destino são endereços *unicast* atribuídos aos *hosts* (ou porta *host*, para ser mais exato). Na Figura 21.1, um pacote *unicast* parte da origem S1 e segue através de roteadores até alcançar o destino D1. Mostramos as redes como *links* entre os roteadores para simplificar a figura.

Note que num **roteamento unicast**, quando um roteador recebe um pacote ele o encaminha através de uma das suas portas, exatamente aquela que leva ao melhor caminho de acordo com a tabela de roteamento. O roteador pode descartar um pacote se ele não for capaz de determinar o endereço de destino na tabela de roteamento dele.

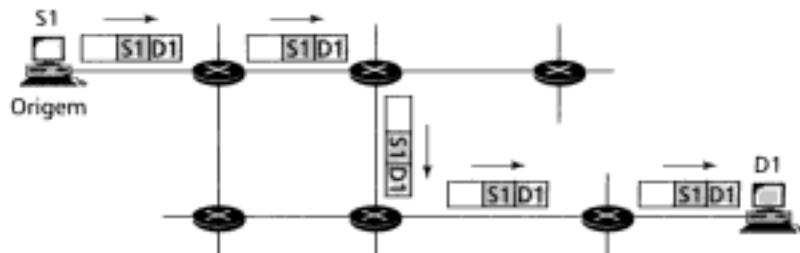


Figura 21.1 Unicasting.

No roteamento *unicast* o roteador encaminha os pacotes recebidos através de uma única porta.

Métrica

Um roteador recebe pacotes de uma rede e os envia às outras redes. Os roteadores geralmente interligam muitas redes diferentes. Quando um roteador recebe um pacote, como ele sabe para que rede enviá-lo? Normalmente, a decisão é tomada com base em algum parâmetro de otimização como, por exemplo, que caminho disponível é a melhor rota? Uma **métrica** é uma espécie de custo de passagem através de uma rede. A métrica total de uma rota particular é igual à soma das métricas das redes que compõem a rota. Um roteador escolhe a rota com a menor métrica total.

A métrica atribuída a cada rede depende do tipo de protocolo. Alguns protocolos de roteamento mais simples, como o **Routing Information Protocol (RIP)**, tratam todas as redes igualmente. O custo de passagem através de cada rede é o mesmo: o número de saltos. Nessa métrica, se um pacote passa através de 10 redes até alcançar o destino, o custo total é 10 saltos.

Outros protocolos, como o **Open Shortest Path First (OSPF)**, permitem que o administrador atribua um custo de passagem através de uma rede baseado no tipo de serviço desejado. Uma rota através de uma rede utilizando OSPF pode ter diferentes métricas. Por exemplo, se o tipo de serviço desejado é o *throughput* máximo, rota (*link*) de satélite pode ser escolhida pois possui uma métrica menor que um *link* de fibra óptica ou outros meios metálicos. Por outro lado, se o tipo de serviço desejado é o atraso mínimo, um *link* de fibra óptica possui uma métrica menor que um *link* de satélite. O protocolo OSPF permite a cada roteador manter muitas tabelas de roteamento baseadas no tipo de serviço desejado.

Em outros protocolos, como o **Border Gateway Protocol (BGP)**, o critério é a política, a qual pode ser configurada pelo administrador. A política define o tipo de rota a ser escolhida.

Roteamento Interno e Externo

Hoje, uma internet pode ser tão grande que um único protocolo de roteamento não consegue controlar sozinho a tarefa de manter atualizadas as tabelas de roteamento de todos os roteadores. Por esta razão, uma internet é dividida em *autonomous systems* (sistemas autônomos). Um **Autonomous System (AS)** é um grupo de redes e roteadores sob uma única autoridade administrativa. O roteamento dentro de um AS é denominado **roteamento interno**. O roteamento entre os AS é denominado **roteamento externo**. Cada AS pode utilizar um protocolo de roteamento interno para controlar o roteamento dentro do AS. Entretanto, inter-AS é permitido um, e somente um, protocolo de roteamento externo.

21.2 PROTOCOLOS DE ROTEAMENTO UNICAST

Existem muitos tipos de protocolos de roteamento internos e externos em uso atualmente nas internets. Nesta seção trataremos apenas os mais comuns. Discutiremos dois protocolos de roteamento interno, RIP e OSPF, e um protocolo de roteamento externo, BGP (veja Figura 21.2).



Figura 21.2 Protocolos de roteamento popular.

Os protocolos RIP e OSPF podem ser utilizados para gerar e atualizar as tabelas de roteamento dinamicamente dentro de um AS, já o BGP é utilizado para geração e atualização dinâmica de tabelas de roteamento entre os roteadores de borda que interligam AS diferentes.

Na Figura 21.3, os roteadores R1, R2, R3 e R4 usam protocolos de roteamento interno e externo. Os outros roteadores usam somente protocolos de roteamento interno. As linhas em traço cheio mostram a comunicação entre roteadores que usam protocolos de roteamento interno. As linhas traçadas mostram a comunicação entre os roteadores que usam um protocolo de roteamento externo.

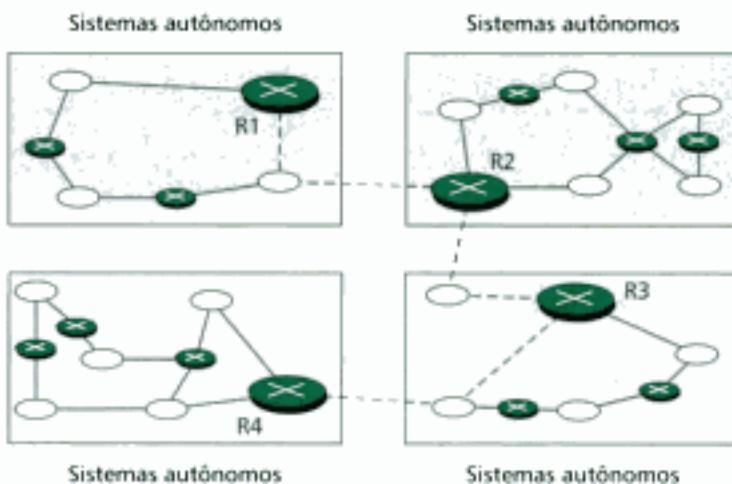


Figura 21.3 Sistemas autônomos.

RIP

O **Routing Information Protocol (RIP)** é um protocolo de roteamento interno usado dentro de um AS. Ele é um protocolo muito simples cujo mecanismo de roteamento está baseado no *vetor de distância* (*distance vector*), o qual baseia-se no algoritmo de Bellman-Ford para calcular e montar as tabelas de roteamento. Nesta seção, estudaremos inicialmente o princípio do vetor de distância aplicado ao RIP. Em seguida, discutiremos o protocolo RIP em si.

Roteamento Baseado no Vetor de Distância

No **roteamento baseado no vetor de distância**, cada roteador compartilha, periodicamente, com os demais roteadores da vizinhança o conhecimento que ele tem sobre a internet conectada através de suas portas. Os três pontos fundamentais sobre o funcionamento do algoritmo são:

- Compartilhamento do conhecimento sobre o AS.** Cada roteador compartilha o conhecimento que possui sobre todo o AS com os vizinhos. No início, o conhecimento que um roteador tem sobre o sistema ao qual está conectado pode ser vago. Entretanto, quanto o roteador sabe inicialmente não é importante, pois ele compartilha toda informação de que dispõe no momento.
- Compartimento com os vizinhos.** Cada roteador transmite o conhecimento sobre as rotas somente aos vizinhos imediatos. Ele transmite todo o conhecimento sobre as rotas que ele tem conhecimento através de todas as interfaces configuradas.
- Compartilhamento em intervalos regulares.** A transmissão do conhecimento sobre as rotas, baseadas nas tabelas, acontece em intervalos de tempo regulares, por exemplo, a cada 30s.

Tabela de Roteamento

Todo roteador mantém uma **tabela de roteamento** contendo entradas para cada rede de destino conhecida. Essas entradas consistem do endereço da rede de destino, da distância mais curta para alcançar esse destino, registrada no contador de saltos, e do endereço do próximo roteador para o qual o pacote deve ser entregue para alcançar o destino final. O contador de saltos revela a quantidade de redes que o pacote encontra no caminho até alcançar o destino final.

A tabela de roteamento pode conter outras informações tais como a máscara de sub-rede ou data de atualização da tabela. A Tabela 21.1 mostra um exemplo de tabela de roteamento.

TABELA 21.1 Tabela de roteamento baseada no vetor de distância

Destino	Contador de saltos	Próximo roteador	Outras informações
163.5.0.0	7	172.6.23.4	
197.5.13.0	5	176.3.6.17	
189.45.0.0	4	200.5.1.6	
115.0.0.0	6	131.4.7.19	

Algoritmo de Atualização RIP

A tabela de roteamento é atualizada mediante a recepção da mensagem de resposta RIP. A seguir podemos ver o algoritmo de atualização usado pelo RIP.

Algoritmo de atualização RIP

Recebe: uma mensagem de resposta RIP

- Incrementa o contador de saltos de uma unidade para cada roteador até o destino.
- Repete os seguintes passos para notificar cada parada do pacote até o destino:
 - Se (o destino não estiver na tabela de roteamento)
 - Adiciona a informação na tabela.
 - Senão
 - Se (o campo próximo salto é o mesmo)
 - Substitui a entrada na tabela pela informação recebida.
 - Senão
 - Se (o valor do contador de saltos for menor que o anterior na tabela)
 - Substitui a entrada na tabela de roteamento.
- Retorna.

Na Figura 21.4, um roteador recebe uma mensagem RIP do roteador C. A mensagem lista as redes de destino e os saltos correspondentes a cada rede. O primeiro passo, de acordo com o algoritmo de atualização, é incrementar 1 no contador de saltos. Em seguida, a mensagem atualizada do pacote RIP e a tabela de roteamento antiga são comparados. O resultado é uma nova tabela de roteamento com o contador de saltos devidamente atualizado para cada rede de destino. Para a Net1 não há nenhuma informação nova, assim a entrada Net1 permanece a mesma.

Para a Net2, a informação na tabela e na mensagem identificam o mesmo valor do próximo salto (roteador C). Embora o valor do contador de saltos na tabela (2) seja menor que o valor da mensagem (5), o algoritmo seleciona o valor recebido na mensagem porque o valor original veio do roteador C. Esse valor torna-se inválido porque o roteador C está informando um novo valor.

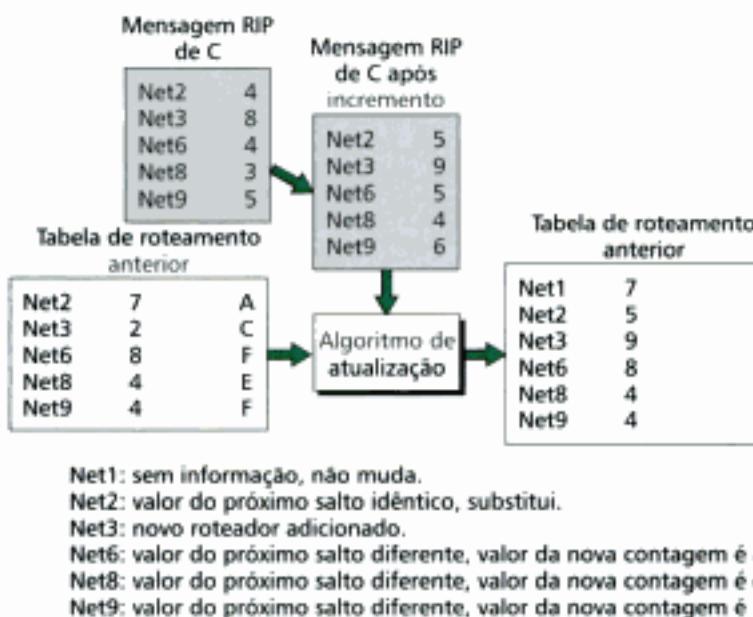


Figura 21.4 Tabelas de roteamento atualizada.

A Net3 é adicionada como uma rota nova. Para a Net6, o pacote RIP possui o contador de saltos menor e isto aparece na nova tabela de roteamento. Tanto a Net8 quanto a Net9 permanecem com os valores originais visto que o contador de saltos correspondente na mensagem não foi modificado.

Inicializando a Tabela de Roteamento

Tão logo um roteador é adicionado à rede, a tabela de roteamento é inicializada através de um arquivo de configuração gravado no próprio roteador. A tabela contém somente as redes diretamente conectadas e os contadores de saltos, os quais são inicializados em 1. O campo próximo salto, o qual identifica o próximo roteador da rede, está vazio. A Figura 21.5 mostra as tabelas de roteamento iniciais em um AS pequeno.

Atualizando a Tabela de Roteamento

Cada tabela de roteamento é atualizada mediante a recepção da mensagem de resposta RIP, seguindo o algoritmo de atualização RIP mostrado anteriormente. A Figura 21.6 mostra as tabelas de roteamento finais para o AS anterior.

OSPF

O protocolo **Open Shortest Path First (OSPF)** é outro protocolo de roteamento interno muito difundido atualmente nas *internetworks*. O domínio de atuação deste protocolo também é um AS. Roteadores especiais denominados **roteadores de borda do AS** são responsáveis pela troca de in-

Hidden page

cionam como áreas secundárias. Entretanto, isto não significa que os roteadores dentro das áreas não possam ser conectados uns aos outros.

Os roteadores dentro da área de *backbone* são denominados **roteadores backbone**. Note que um roteador *backbone* também pode ser um roteador de borda de área.

Se, devido a algum problema, a conectividade entre um *backbone* e uma área é quebrada, um **link virtual** entre os roteadores pode ser criado pelo administrador para permitir continuidade das funções do *backbone* com a área principal.

Além disso, cada área possui uma identificação. A identificação da área do *backbone* é a zero. A Figura 21.7 mostra um AS e as respectivas áreas internas.

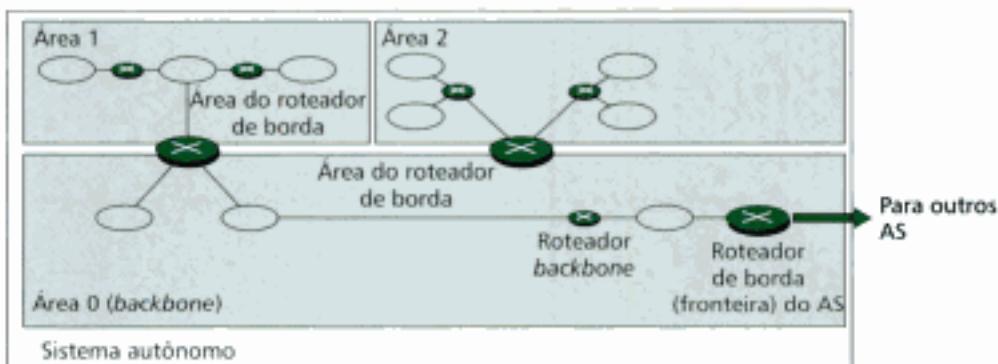


Figura 21.7 Áreas em um sistema autônomo.

Métrica

O protocolo OSPF permite ao administrador atribuir um custo, denominado **métrica**, a cada roteador. A métrica pode ser baseada em um tipo de serviço (atraso mínimo, throughput máximo, etc.). O fato é que, um roteador em geral tem tabelas de roteamento múltiplas, baseadas possivelmente em muitos tipos de serviços diferentes.

Roteamento Baseado no Estado de Link

O protocolo OSPF usa o **roteamento baseado no link state (estado de link)** para atualizar as tabelas de roteamento numa área. Antes de discutirmos os detalhes do protocolo OSPF, vamos examinar o roteamento baseado no *link state*, um processo pelo qual cada roteador compartilha o conhecimento sobre a vizinhança com os demais roteadores da área. Os três pontos fundamentais sobre o funcionamento deste tipo de roteamento são:

- 1. Compartilhamento do conhecimento sobre a vizinhança.** Cada roteador envia uma mensagem contendo o *estado da vizinhança* para os outros roteadores da área.
- 2. Compartilhamento com os demais roteadores.** Cada roteador envia uma mensagem contendo o estado da vizinhança para *todos os roteadores na área*. Isto é conhecido por **inundação (flooding)**, um processo por meio do qual um roteador envia a informação de que tem conhecimento para todos os vizinhos (através de todas as portas de saída de que dispõe). Cada vizinho envia um pacote aos demais e assim continua o processo. O roteador que recebe um pacote envia uma cópia aos demais vizinhos. Eventualmente, um roteador (sem exceção) recebe uma cópia da mesma informação.
- 3. Compartilhamento quando há modificações na rede.** Cada roteador compartilha o estado da vizinhança somente quando detecta alguma modificação nela. Esta regra contrasta com a regra do roteamento baseado no vetor de distância, onde a informação é enviada aos outros roteadores em intervalos regulares de tempo, independentemente da mudança. Esta característica resulta em um tráfego menos intenso na internet, se comparado ao roteamento baseado no vetor de distâncias.

A idéia acerca do roteamento baseado no *link state* é que cada roteador deve possuir uma descrição exata da topologia da internet a cada momento. Assim, os roteadores tiram um "retrato" fidedigno da internet a qual estão conectados. Nesta topologia, um roteador pode calcular a rota mais curta entre ele e a rede de interesse. A topologia representada aqui significa um gráfico contendo nós e conexões. Porém, para representar uma internet através de um gráfico ainda precisamos de mais definições.

Tipos de *Links*

Na terminologia OSPF, uma conexão é chamada de *link*. Quatro tipos de *links* OSPF foram definidos: ponto a ponto, transitório, *stub* e virtual (veja a Figura 21.8).



Figura 21.8 Tipos de *links*.

Link Ponto a Ponto Um *link ponto a ponto* conecta dois roteadores diretamente, isto é, sem a existência de qualquer outro *host* ou roteador entre eles*. Dito de outra forma, o propósito do *link* (rede) ponto a ponto é apenas conectar os dois roteadores. Um exemplo deste tipo de *link* é a conexão de dois roteadores via linha telefônica ou uma linha E. Não é necessário atribuir um endereço de rede para este tipo de *link*. Graficamente, os roteadores são representados pelos nós e os *links* representam a conexão bidirecional interligando tais nós. As métricas, que possivelmente são as mesmas, são mostradas nas duas extremidades em cada lado da conexão. Assim, um roteador possui um único vizinho do outro lado do *link* (veja a Figura 21.9).

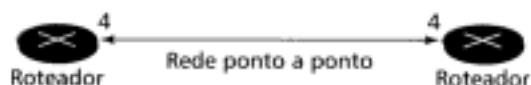


Figura 21.9 Link ponto a ponto.

Link Transitório Um *link transitório* é uma rede com muitos roteadores ligados a ela. Os dados podem entrar ou sair através de qualquer um dos roteadores. Todas as LANs e algumas WANs com dois ou três roteadores são deste tipo. Neste caso, cada roteador possui geralmente múltiplos vizinhos. Por exemplo, considere a rede Ethernet da Figura 21.10a. O roteador A possui os roteadores vizinhos B, C, D e E. O roteador B possui os roteadores vizinhos A, C, D e E. Se quisermos mostrar o relacionamento entre a vizinhança nesta situação podemos usar o gráfico da Figura 21.10b.

Esta representação não é nem eficiente nem realista. Ela não é eficiente porque cada roteador precisa notificar a vizinhança a respeito dos outros quatro roteadores, totalizando 20 notificações. Ela é irreal porque não existe um *link* único entre cada par de roteadores; há somente uma rede que serve como cruzamento de informação entre todos os cinco roteadores.

Para mostrar que cada roteador é conectado aos demais através de uma única rede, a rede em si é representada por um nó. Entretanto, como a rede não possui funcionalidade eletrônica, ela não substitui um roteador. Um dos roteadores da rede é eleito, adquirindo tal responsabilidade. Esse roteador desempenha duplo papel: funcionar como um roteador de fato e ser o roteador designado da rede. Podemos utilizar a representação da Figura 21.10c para mostrar as conexões de uma rede transitória.

* N. de R. T.: Neste ponto o autor fez uma distinção entre *host* (computador) e roteador. Entretanto, em geral, os roteadores são tratados como *hosts* da rede.

Hidden page

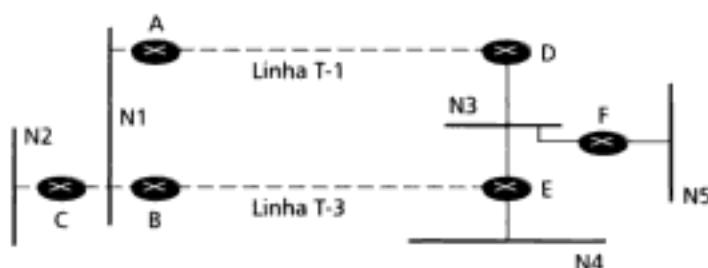


Figura 21.12 Exemplo de uma internet.

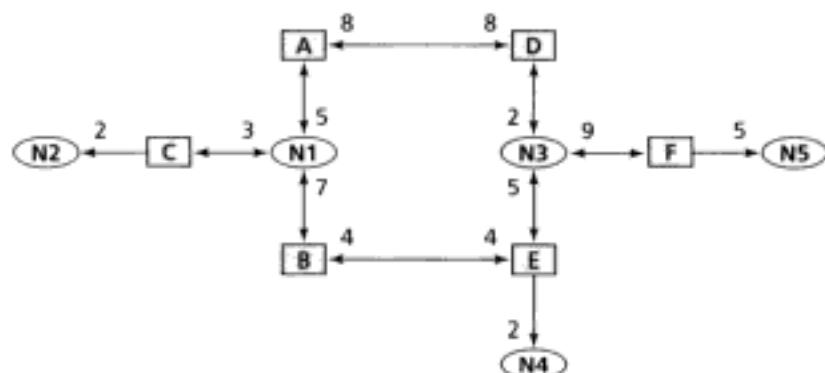


Figura 21.13 Representação gráfica de uma internet.

LSAs

Para compartilhar informação sobre a vizinhança, cada entidade da rede distribui **LSAs (Link State Advertisements)**. Um LSA comunica os estados das entidades ligadas aos *links*. Podemos definir cinco tipos diferentes de LSAs, dependendo do tipo de entidade envolvida (veja a Figura 21.14).



Figura 21.14 Tipos de LSAs.

Link com o Roteador Uma notificação de *link* com o roteador define os *links* de um roteador verdadeiro. Um roteador usa este tipo de notificação para divulgar informação sobre todos os *links* dele e o que está do outro lado do *link* (vizinhos). Observe a representação de um *link* com o roteador na Figura 21.15.

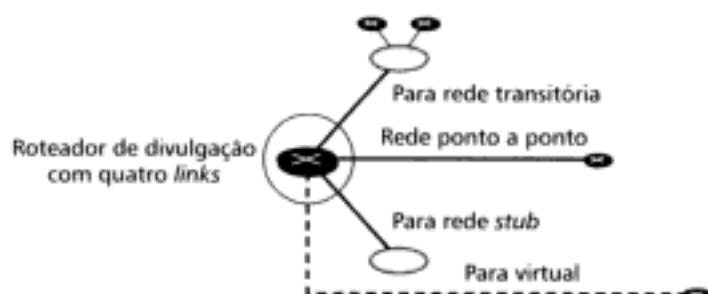


Figura 21.15 Link com o roteador.

Link com a Rede Uma notificação do *link* com a rede define os *links* de uma rede. Um roteador designado da rede transitória distribui este tipo de pacote LSA. O pacote notifica a existência de todos os roteadores conectados à rede (veja a Figura 21.16).



Figura 21.16 Link rede.

Link Direto com a Rede As notificações dos *links* com o roteador e com a rede inundam a área com informação sobre os roteadores envolvidos dentro de uma área. Contudo, um roteador também deve saber sobre as redes fora da área de residência. Os roteadores de borda podem proporcionar esta informação. Um roteador de borda de área está ativo em mais de uma área do AS. Ele recebe as notificações do *link* com o roteador e do *link* com a rede e cria uma tabela de roteamento para cada área. Por exemplo, na Figura 21.17, o roteador R1 é um roteador de borda da área 0. Ele possui duas tabelas de roteamento, uma para área 0 e outra para a área 1. O roteador R1 inunda a área 1 com informações sobre como alcançar uma rede localizada na área 0. Da mesma forma, o roteador R2 inunda a área 2 com informações sobre como alcançar a mesma rede na área 0.

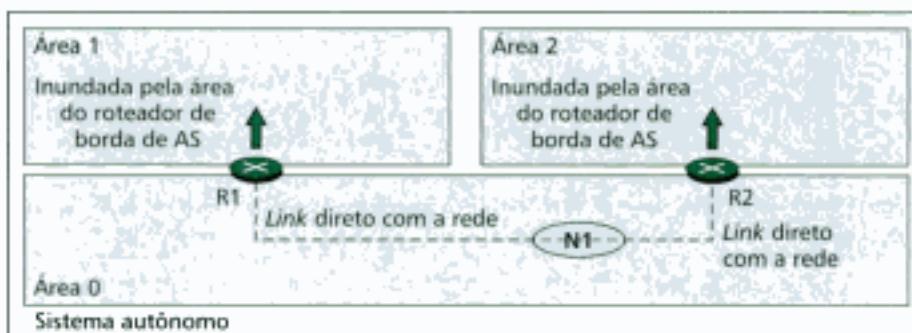


Figura 21.17 Link direto com a rede.

Link Direto com o Roteador de Borda do AS A notificação transmitida no *link* direto com a rede informa aos roteadores os custos para alcançar todas as redes dentro de um AS. Entretanto, como saber sobre os custos para alcançar as redes externas ao AS? Se um roteador residente numa área quiser enviar um pacote para fora do AS, primeiramente deve conhecer uma rota até um roteador de borda do AS. O *link* direto com o roteador de borda do AS proporciona este tipo de informação. Os roteadores de borda de área inundam as respectivas áreas com tais informações (veja a Figura 21.18).

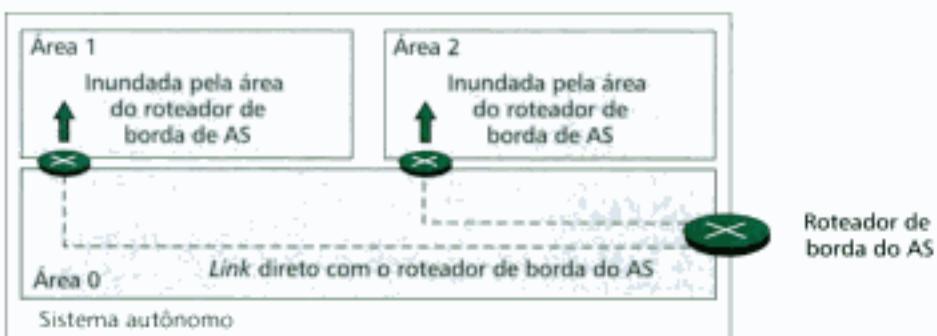


Figura 21.18 Link direto com o roteador de borda do AS.

Link Externo Embora a notificação enviada no *link* anterior possibilite a cada roteador saber uma rota até um roteador de borda do AS, esta informação ainda não é suficiente. Um roteador interno ao AS muitas vezes deseja saber de fato qual a disponibilidade das redes externas ao AS. Esta é a razão de ser do *link* externo. O roteador de borda do AS inunda a área interna do AS com os custos para cada rede externa. Para isto, o roteador usa uma tabela de roteamento criada por um protocolo de roteamento externo. Além do mais, cada pacote de notificação informa sobre uma única rede. Se houver mais de uma rede são enviadas notificações separadas. A Figura 21.19 ilustra um *link* externo.

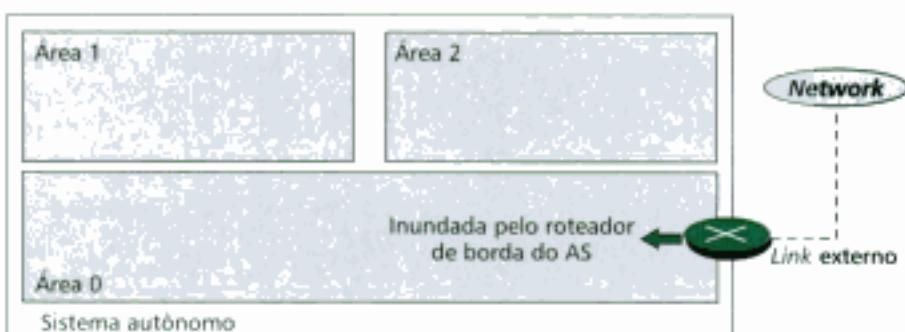


Figura 21.19 Link externo.

Base de dados de *Link State*

Todo roteador numa área de um AS recebe os *links* com o roteador e com a rede (LSAs) dos outros roteadores e formam uma **base de dados de *link state***. Perceba que os roteadores pertencentes a uma mesma área possuem a mesma base de dados de *link state*.

Uma base de dados de *link state* é uma representação tabular da topologia da internet dentro da área. Ela mostra o relacionamento entre cada roteador e os vizinhos, incluindo as métricas.

No OSPF, todos os roteadores têm a mesma base de dados de *link state*.

Algoritmo de Dijkstra

Para calcular esta tabela de roteamento, cada roteador aplica o algoritmo de Dijkstra à base de dados de *link state*. O **algoritmo de Dijkstra** determina a menor distância entre duas redes usando um gráfico formado de nós e conexões. O algoritmo divide os nós em dois conjuntos: nó tentativa e nó permanente. Ele escolhe um nó tentativa, examina-o e, se passar pelo critério, o nó é transformado em nó permanente. Informalmente, podemos definir o algoritmo de Dijkstra usando a sequência lógica a seguir:

Algoritmo de Dijkstra

1. Inicia com o nó local (roteador): a raiz da árvore.
2. Atribui o custo 0 a este nó e o transforma no primeiro nó permanente.
3. Examina cada nó da vizinhança do nó inicial (transformado em permanente).
4. Atribui um custo cumulativo a cada nó e os transforma em tentativa.
5. Na lista de nós tentativas
 1. Encontra o nó com o menor custo cumulativo e o transforma em permanente.
 2. Se um nó puder ser alcançado em mais de uma direção.
 1. Seleciona a direção com o menor custo cumulativo.
6. Repete os passos 3 a 5 até que cada nó seja transformado em permanente.

A Figura 21.20 ilustra algumas etapas do algoritmo de Dijkstra aplicadas ao nó A da internet exemplo (Figura 21.13). Os números inteiros em cada nó representam o custo cumulativo a partir do nó raiz. Perceba que, se uma rede puder ser alcançada através de duas direções com dois custos cumulativos diferentes, a direção com o menor custo é mantida e a outra é descartada.

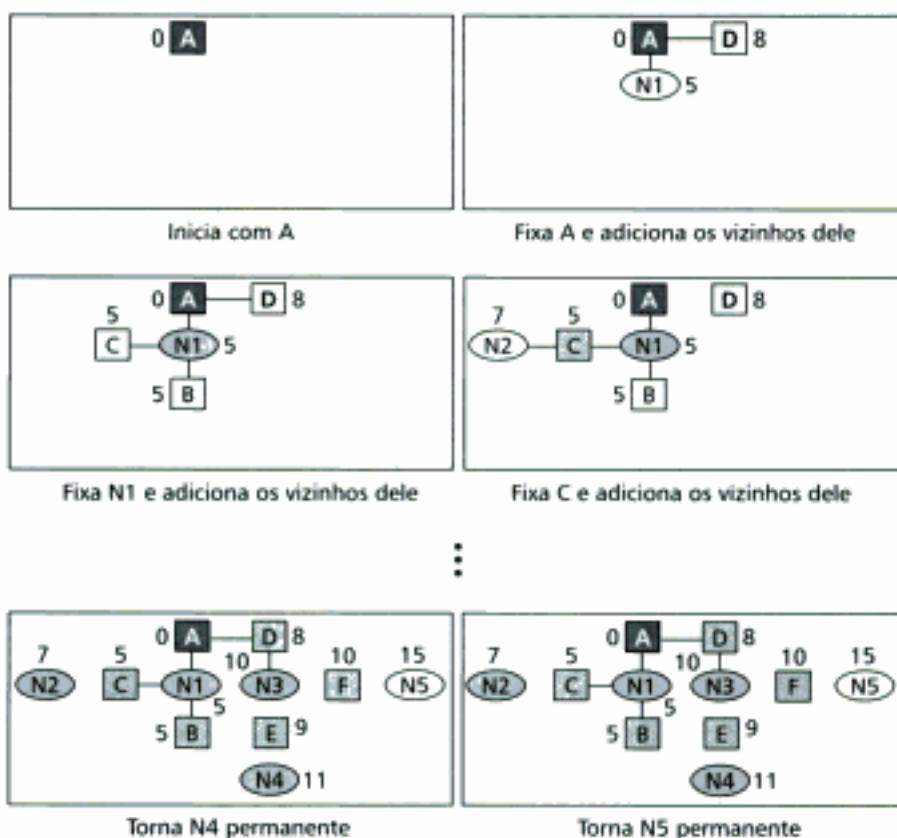


Figura 21.20 Cálculo da rota mais curta.

Tabela de Roteamento

Cada roteador usa o método da árvore de menor caminho para montar a respectiva tabela de roteamento. A tabela de roteamento mostra o custo para alcançar as redes dentro da área. Os roteadores fazem a determinação do custo para as redes fora da área usando mensagens de notificação de *link* direto com a rede, o *link* direto com o roteador de borda e o *link* externo. A Tabela 21.2 ilustra a tabela de roteamento para o roteador A.

TABELA 21.2 Tabela de roteamento de estado do *link* para o roteador A

Rede	Custo	Próximo roteador	Outras informações
N1	5		
N2	7	C	
N3	10	D	
N4	11	B	
N5	15	D	

BGP

O protocolo BGP (Border Gateway Protocol) é um **protocolo de roteamento entre sistemas autônomos**. Este protocolo apareceu pela primeira vez em 1989 e a versão atual é a BGP4. O BGP baseia-se num método de roteamento denominado **vetor de caminhos (path vector)**. Entretan-

to, antes de iniciarmos a descrição do roteamento baseado no vetor de caminhos, vejamos por que os dois métodos discutidos anteriormente – roteamento do vetor de distâncias e roteamento baseado no *link state* – não são bons candidatos para o roteamento envolvendo sistemas autônomos diferentes.

O método baseado no vetor de distâncias não é um bom candidato porque há ocasiões aonde o roteador com o menor valor no contador de saltos não é a rota preferida. Por exemplo, pode não ser desejável que um pacote passe através de um AS inseguro, mesmo que esta seja a rota mais curta. Além disso, o roteamento baseado no vetor de distância é instável visto que os roteadores anunciam somente o número de saltos até o destino, sem definir realmente que caminho conduz ao destino. Um roteador que recebe um pacote de notificação do vetor de distâncias pode ser ludibriado se o caminho mais curto contiver o roteador que estiver recebendo e processando o pacote.

O roteamento *link state* não é uma boa solução para o roteamento entre sistemas autônomos diferentes porque uma internet pode ser de fato muito grande para este tipo de roteamento. Para usar o roteamento *link state* na maioria das internets é necessário que cada roteador da rede mantenha uma base de dados monstruosa de *link state*. Outro fator que pesa é o tempo gasto excessivamente na determinação da tabela de roteamento baseada no algoritmo de Dijkstra.

Roteamento Baseado no Vetor de Caminhos

O método de roteamento baseado no vetor de caminhos é diferente tanto do roteamento baseado no vetor de distâncias quanto do roteamento *link state*. Toda entrada na tabela de roteamento contém a rede de destino, o próximo salto (roteador) e o caminho até alcançar o destino. O caminho é definido usualmente como uma lista ordenada de ASs através dos quais o pacote deve viajar até atingir o destino. A Tabela 21.3 ilustra um exemplo de tabela de roteamento baseada no vetor de caminhos.

TABELA 21.3 Tabela de roteamento baseada no vetor de caminhos

Rede	Próximo roteador	Caminho
N01	R01	AS14, AS23, AS67
N02	R05	AS22, AS67, AS05, AS89
N03	R06	AS67, AS89, AS09, AS34
N04	R12	AS62, AS02, AS09

Mensagem Vetor de Caminhos

Os roteadores de borda do AS que participam do roteamento do vetor de caminhos notificam a acessibilidade das redes nos próprios domínios de AS aos demais roteadores de borda dos ASs da vizinhança. O conceito de vizinhança aqui é o mesmo descrito para os protocolos RIP e OSPF. Dois roteadores de borda conectados à mesma rede são vizinhos.

Devemos mencionar ainda que o roteador de borda do AS recebe a informação do sistema interno através de um algoritmo de roteamento tal como o RIP ou OSPF.

Todo roteador que receber uma mensagem vetor de caminhos verifica se o caminho notificado concorda com a política estabelecida para o roteador (política = conjunto de regras impostas pelo administrador que controla as rotas). Caso concorde, o roteador atualiza a tabela de roteamento e modifica a mensagem antes de retransmiti-la ao próximo vizinho. As modificações efetuadas pelo roteador consistem em adicionar as informações relativas a ele, isto é, inserir o número do AS ao caminho e substituir a entrada do próximo roteador pela própria identificação.

Por exemplo, a Figura 21.21 mostra uma internet constituída de quatro sistemas autônomos. O roteador R1 envia uma mensagem vetor de caminhos notificando a acessibilidade de N1. O roteador R2 recebe a mensagem, atualiza a tabela de roteamento dele e, após adicionar a informação sobre o AS dele ao caminho e inserir-se na tabela como próximo roteador, envia a mensagem ao roteador R3. O roteador R3 recebe a mensagem, atualiza a tabela de roteamento dele e envia a mensagem, após as modificações, ao roteador R4.

Hidden page

Tipos de Mensagem

O BGP usa quatro tipos diferentes de mensagens: *open*, *update*, *keep-alive* e *notification* (veja a Figura 21.22).

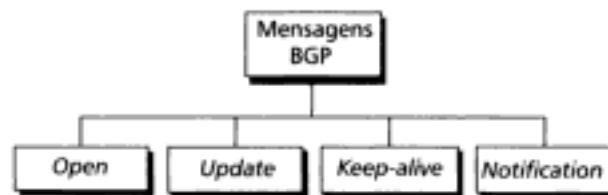


Figura 21.22 Tipos de mensagens BGP.

Mensagem Open Supondo que um roteador esteja utilizando o BGP, toda vez que ele deseja estabelecer um relacionamento com a vizinhança é aberta uma conexão com um vizinho e é enviada uma **mensagem open**. Se o vizinho aceitar o relacionamento com a vizinhança, ele responde com a mensagem *keep-alive*, significando que o relacionamento foi aceito entre os dois roteadores.

Mensagem Update A **mensagem update** é o coração do protocolo BGP. Ela é utilizada por um roteador para remover destinos previamente notificados, anunciar uma rota para um novo destino ou para ambos. É importante ressaltar que o BGP pode remover muitos destinos já notificados de uma só vez, mas ele só consegue notificar um novo destino por mensagem.

Mensagem Keep-Alive Os roteadores que estiverem rodando o protocolo BGP trocam **mensagens keep-alive** regularmente para comunicar que eles estão “vivos” (ativos).

Mensagem Notification A **mensagem notification** é enviada por um roteador sempre que uma condição de erro é detectada ou um roteador quer terminar uma conexão.

21.3 ROTEAMENTO MULTICAST

No **roteamento multicast** há uma origem e vários destinos. Dizemos que o relacionamento é de um para vários. Neste tipo de comunicação, a origem possui um endereço *unicast*, mas os endereços de destino são *multicast* (classe D). O endereço de grupo define os membros do grupo. A Figura 21.23 mostra a ideia central do roteamento *multicast*. Um pacote *multicast* é enviado da origem S1 destinada ao grupo G1.

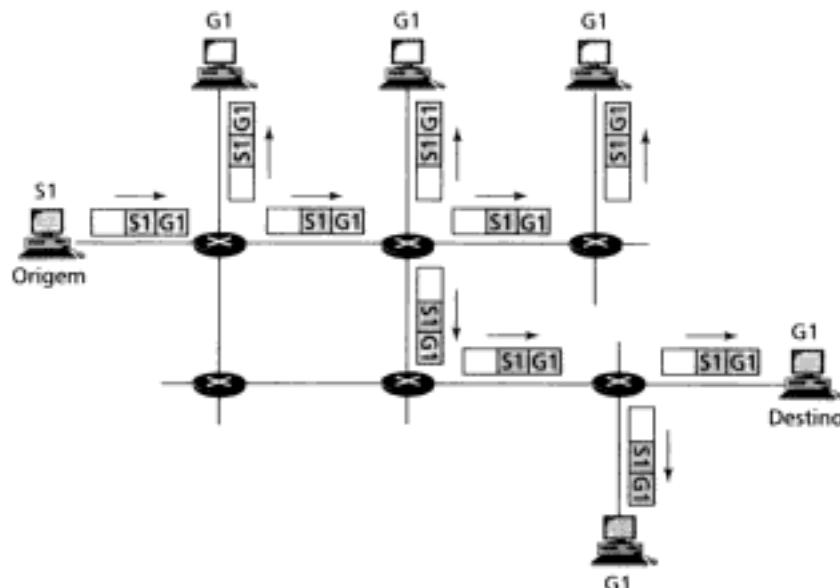


Figura 21.23 Multicasting.

No roteamento *multicast* quando um roteador recebe um pacote ele pode encaminhá-lo através de muitas portas. O roteador pode descartar um pacote se ele não estiver no caminho de *multicast*.

No roteamento *multicast*, um roteador pode encaminhar um pacote recebido através de muitas portas.

Broadcasting é um caso especial do *multicasting*, onde o grupo contém todos os *hosts* da rede. A Internet não suporta *broadcasting* explicitamente devido ao enorme tráfego criado na rede e a alta demanda por largura de banda. Imagine o tráfego gerado na Internet se 1000 pessoas enviassem, ao mesmo tempo, uma mensagem para cada uma das outras pessoas conectadas à Internet! Entretanto, como veremos em breve, o *broadcasting* é usado implicitamente como um pré-lúdio ao *multicasting*.

Outro termo bastante utilizado no roteamento *multicast* é **inundação (flooding)**. A inundação relaciona-se tanto ao *multicasting* quanto ao *broadcasting*. Durante um processo de inundação, um roteador encaminha um pacote para fora de todas as portas, exceto pela porta por onde o pacote foi recebido. O processo de inundação proporciona *broadcasting*, mas também cria *loops*. Um roteador recebe sempre o mesmo pacote por portas diferentes, uma após a outra. Muitas cópias do mesmo pacote circulam e congestionam o tráfego (gerando *jams*).

Nesta seção, iniciaremos a discussão com o IGMP, um protocolo da camada de rede responsável pelo gerenciamento de grupos.

IGMP

O **IGMP (Internet Group Management Protocol)** é um dos protocolos necessários, mas não suficiente (como veremos), ao processo de *multicasting*.

Gerenciamento de Grupos

Para operacionalizar o *multicasting* na Internet precisamos de roteadores capazes de rotear pacotes *multicast*. As tabelas de roteamento destes roteadores devem ser atualizadas usando um dos protocolos de roteamento *multicast* que veremos mais adiante.

O IGMP não é um protocolo de roteamento *multicast*, ele é um protocolo que gerencia membros de um grupo. Hoje, em muitas redes há pelo menos um **roteador *multicast*** que distribui os pacotes *multicast* para *hosts* e/ou outros roteadores. O IGMP fornece informação de suporte aos roteadores *multicast* informando sobre o *status* dos membros (*hosts* ou roteadores) conectados à rede.

Um roteador *multicast* pode receber milhares de pacotes *multicast* por dia oriundos de grupos diferentes. Se um roteador não tiver nenhum conhecimento antecipado do *status* dos *hosts*, ele é obrigado a transmitir todos os pacotes em caráter de *broadcast* para a rede. Isto cria muito tráfego e consome muita largura de banda do *link*. Uma solução melhor é manter uma lista de grupos na rede para o qual há pelo menos um membro interessado no *multicast*. O IGMP auxilia o roteador *multicast* a criar e a manter atualizada esta lista.

IGMP é um protocolo para gerenciamento de grupos. Ele auxilia um roteador *multicast* a criar e a manter atualizada a lista de membros interessados no *multicast* relacionados a cada interface do roteador.

Mensagens

O IGMP possui três versões, sendo a atual a IGMPv2. O protocolo IGMPv2 possui três tipos de mensagens: **mensagem de consulta (query)**, **relatório de assinaturas de membros** e **relatório de saídas (leave)**. Há dois tipos de mensagens de consulta: a geral e a especial.

Formato da Mensagem

A Figura 21.25 mostra o formato padrão da mensagem IGMPv2.

- **Tipo.** Este campo de 8-bits define o tipo de mensagem, conforme está ilustrado na Tabela 21.4. O valor do campo tipo é mostrado tanto em hexadecimal quanto em binário.

Hidden page



Figura 21.26 Operação IGMP.

uma lista de ID de grupos e retransmitem a mensagem de interesse a um grupo específico ao roteador responsável pela distribuição.

Por exemplo, na Figura 21.26, o roteador R é um roteador de distribuição. Porém, existem outros dois roteadores *multicast* (R1 e R2) os quais, dependendo da lista de grupos mantida pelo roteador R, podem receber do roteador R nesta rede. Os roteadores R1 e R2 podem ainda ser distribuidores para alguns dos grupos nas outras redes, mas não na rede mostrada.

Entrando (*Joining*) no Grupo Tanto um *host* quanto um roteador podem entrar (*join*) no grupo. Um *host* mantém uma lista de processos assinando um grupo qualquer. Quando um processo deseja entrar para um novo grupo, ele transmite uma solicitação ao *host*. O *host* adiciona o nome do processo e o nome do grupo solicitado à lista dele. Se esta for a primeira entrada para este grupo particular, o *host* envia um relatório para o novo grupo assinado. Caso não seja, não há necessidade do relatório de assinaturas para o grupo, visto que o *host* já é membro do grupo (isto é, ele já recebe pacotes *multicast* endereçados a este grupo).

Um roteador também mantém uma lista de IDs de grupos que mostram as redes assinadas conectadas em cada interface. Quando há um novo interessado nos grupos de quaisquer interfaces, o roteador envia um relatório de assinaturas. Assim, nesse caso, o roteador age como um *host*, mas a lista de grupos dele é muito maior porque ela acumula todos os membros leais que estão conectados nas interfaces do roteador. Note que o relatório de assinaturas é enviado através de todas as interfaces do roteador, exceto aquela por onde o novo interessado chega. A Figura 21.27 ilustra um relatório de assinaturas de membros enviado por um *host* ou um roteador.

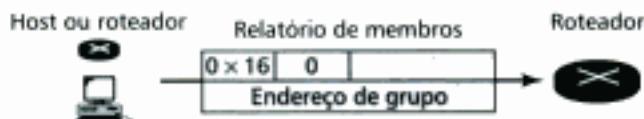


Figura 21.27 Relatório de assinaturas de membros.

O protocolo requer que o relatório de assinaturas seja enviado duas vezes, um após o outro, em intervalos de tempo curtos. Essa redundância assegura que, se o primeiro relatório se perder ou chegar danificado, o segundo poderá substituí-lo.

No IGMP, um relatório de assinaturas de membros é enviado duas vezes, um após o outro.

Saindo (*Leaving*) do Grupo Quando um *host* percebe que nenhum processo está interessado num grupo específico dele, ele envia um relatório de saída (*leave*). Similarmente, quando um roteador percebe que nenhuma das redes conectadas às interfaces dele está interessada num grupo específico, ele envia um relatório de saída desse grupo.

Hidden page

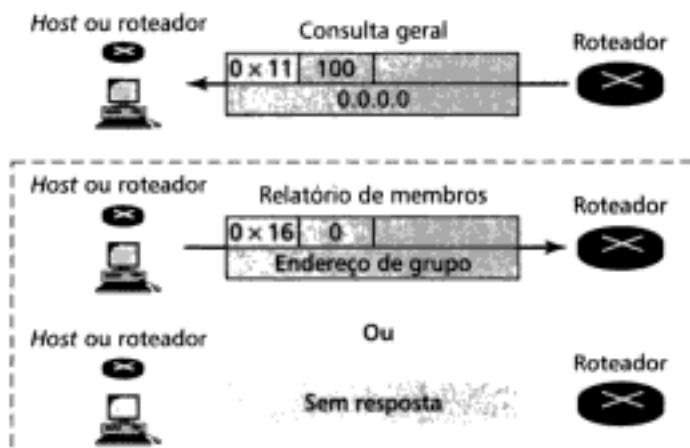


Figura 21.29 Mensagem de consulta (query) geral.

Atraso de Resposta

Como foi dito antes, para evitar tráfego desnecessário, o IGMP usa uma **estratégia supressão com atraso aleatório**. Quando um *host* ou roteador recebe uma mensagem de consulta, não responde imediatamente, atrasando a resposta. Todo *host* ou roteador usa um número aleatório para criar um temporizador que expira num tempo entre 1 e 10s. O tempo de expiração pode acontecer em passos de 1s ou menos. Um temporizador é configurado para cada grupo da lista. Por exemplo, um temporizador para o primeiro grupo pode expirar em 2s, mas o temporizador do terceiro grupo expira em 5s. Cada *host* ou roteador espera até que o temporizador dele expire, antes de enviar um relatório de assinaturas. Durante este tempo de espera, se o temporizador de outro *host* ou roteador expirar, para o mesmo grupo, esse *host* ou roteador envia um relatório de assinaturas. Visto que, como vemos, o relatório acontece em modo *broadcast*, o *host* ou roteador que estiver aguardando receber o relatório e toma conhecimento de que não é necessário enviar um relatório duplicado para este grupo. Assim, a estação que estiver esperando reseta o temporizador dela.

Exemplo 1

Imagine que existam três *hosts* numa rede, conforme ilustra a Figura 21.30. Uma mensagem de consulta foi recebida no tempo $t = 0$; o tempo de atraso aleatório (em décimos de segundos) para cada grupo é mostrado próximo ao grupo. Apresente a seqüência de mensagens de relatório.

Solução

Os eventos de geração das mensagens ocorrem nesta seqüência:

1. **Tempo 12.** O temporizador para o endereço 228.42.0.0 no *host* A expira e um relatório de assinatura é enviado. O relatório é recebido pelo roteador e cada *host* da rede, incluindo o *host* B, reseta o temporizador para o endereço 228.42.0.0.
2. **Tempo 30.** O temporizador para o endereço 225.14.0.0 no *host* A expira e um relatório de assinatura é enviado. O relatório é recebido pelo roteador e cada *host* da rede, incluindo o *host* C, reseta o temporizador para o endereço 225.14.0.0.
3. **Tempo 50.** O temporizador para o endereço 251.71.0.0 no *host* B expira e um relatório de assinatura é enviado, o qual é recebido pelo roteador e por cada *host*.
4. **Tempo 70.** O temporizador para o endereço 230.43.0.0 no *host* C expira e um relatório de assinatura é enviado. O relatório é recebido pelo roteador e cada *host* da rede, incluindo o *host* A, reseta o temporizador para o endereço 230.43.0.0.

Observe que, se cada *host* tivesse enviado um relatório para cada grupo da lista, teríamos sete relatórios. Usando a estratégia de supressão foram enviados apenas quatro relatórios.

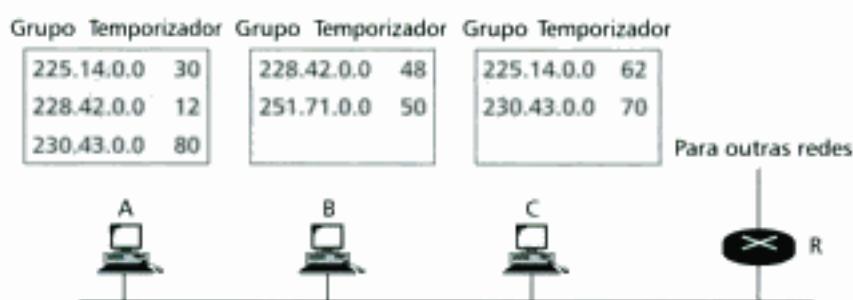


Figura 21.30 Exemplo 1.

Roteador Query

Mensagens de consulta (*query*) podem gerar muitas respostas. Para evitar o tráfego desnecessário, o IGMP elege um roteador como o **roteador query** para cada rede*. Assim, o único roteador autorizado a enviar mensagens de consulta na rede será este roteador designado. Os outros roteadores são passivos, isto é, eles recebem as respostas e atualizam as respectivas listas.

Árvores de Multicast

Os objetivos do *multicasting* são os seguintes:

- Cada membro de um grupo deve receber uma, e somente uma, cópia do pacote *multicast*. A recepção de múltiplos pacotes não é permitida.
- As entidades não-membros dos grupos não podem receber uma cópia.
- Não deve haver a formação de *loops* no roteamento, isto é, um pacote não deve visitar um roteador mais de uma vez.
- A rota da origem para cada destino deve ser ótima (o menor caminho).

Tais objetivos podem ser atingidos usando o algoritmo de *spanning tree* (discutido no Capítulo 16). Dois tipos de árvores são usadas para *multicasting*: SBT (Source-Based Tree – Árvore Multicast Baseada na Fonte) e GST (Group-Shared Tree – Árvore Multicast Compartilhada).

Árvore Multicast Baseada na Fonte – SBT

No método da **árvore baseada na fonte**, uma única árvore é formada para cada fonte que transmite a um grupo. Assim, a formação da árvore baseia-se tanto na fonte quanto no grupo. Se houver N grupos e M fontes diferentes no sistema, pode haver um máximo de $N \times M$ árvores diferentes, uma para cada combinação fonte-grupo. Por exemplo, se num determinado momento, uma fonte (origem) deseja enviar um pacote *multicast* para um grupo com um endereço classe D 228.9.28.40 é formada uma árvore correspondente para esta finalidade. Se após 2 minutos a fonte desejar enviar outro pacote *multicast* para o grupo 230.6.4.2 (um grupo diferente), a árvore é modificada. Na abordagem SBT, a combinação fonte-grupo determina a estrutura da árvore.

Na abordagem SBT (Source-Based Tree – árvore baseada na fonte), a combinação fonte – grupo determina a estrutura da árvore.

Duas abordagens foram usadas para criar e otimizar uma árvore *multicast* baseada na fonte. A primeira abordagem, usada no DVMRP, é uma extensão do roteamento *unicast* baseado no vetor de distâncias (tal como o RIP). A segunda abordagem, usada no MOSPF, é uma exten-

* N. de R. T.: Em geral, o roteador com menor endereço IP é eleito como o roteador *query* da rede.

são do roteamento *unicast* baseado no *link state* (tal como o OSPF). Outro protocolo, PIM-DM, usa o RIP ou o OSPF, dependendo da necessidade. Todas essas abordagens serão discutidas na próxima seção.

Árvore Multicast Compartilhada – GST

No método da **árvore multicast compartilhada**, cada grupo do sistema compartilha a mesma árvore. Se houver N grupos no sistema como um todo, poderá haver no máximo N árvores, uma para cada grupo. Por exemplo, se num determinado momento, uma fonte necessita enviar um pacote *multicast* a um grupo com endereço classe D 226.7.18.10, é formada uma árvore correspondente para este propósito. Se, alguns segundos depois, outra fonte precisa enviar outro pacote ao mesmo grupo, a árvore será a mesma. Mas se a fonte anterior, ou qualquer outra, precisa enviar um pacote para o grupo 229.5.80.10, uma nova árvore será formada. Assim, as árvores mudam quando os grupos mudam e permanecem as mesmas para o grupo produzido a partir da fonte. Na abordagem GST o grupo determina a estrutura da árvore.

Na abordagem GST (Group-Shared Tree – árvore *multicast* compartilhada), o grupo determina a estrutura da árvore.

Este método também possui duas abordagens para determinar a árvore *multicast*: árvore de Steiner e a árvore **rendezvous-point**. Discutiremos a árvore baseada num ponto central (árvore *rendezvous-point*) quando introduzirmos os protocolos CBT e PIM-SP (discutidos na próxima seção). A árvore de Steiner é apenas teórica e ainda não foi implementada.

MBONE

As comunicações de multimídia e em tempo real têm feito crescer a necessidade por *multicasting* na Internet. Contudo, apenas uma pequena fração dos roteadores da Internet são roteadores *multicast*. Assim, um roteador *multicast* pode não conseguir encontrar outro roteador *multicast* na vizinhança para encaminhar o pacote *multicast*. Embora este problema seja resolvido nos próximos anos, através da adição de novos roteadores *multicast* nas redes, há uma solução alternativa para este problema. A solução é o **tunelamento**. Os roteadores *multicast* são vistos como um grupo de roteadores no topo dos roteadores *unicast*. Os roteadores *multicast* não podem ser conectados fisicamente, pois inexiste a vizinhança, mas podem ser conectados logicamente uns com os outros. A Figura 21.31 ilustra essa idéia. Nessa figura, somente os roteadores marcados com círculos traçados são capazes de realizar *multicasting*. Sem o esquema de tunelamento estes roteadores estão isolados, formando ilhas. Para habilitar o *multicasting*, construímos um **Multicast Backbone (MBONE)** fora destes roteadores isolados usando o conceito de tunelamento.

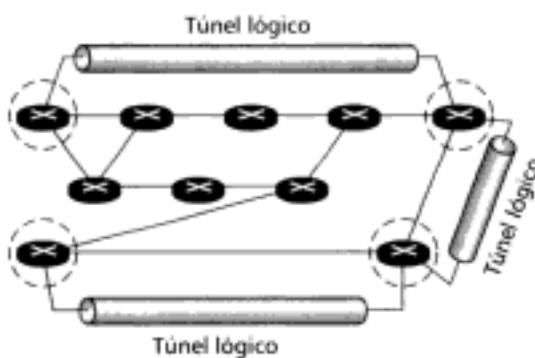


Figura 21.31 Tunelamento lógico.

Um **túnel lógico** é estabelecido encapsulando o pacote *multicast* dentro de um pacote *unicast*. O pacote *multicast* torna-se o *payload* (dados) do pacote *unicast*. Os roteadores intermediários (não *multicast*) roteiam os pacotes *unicast* para os roteadores *multicast*. Este esquema funcio-

Hidden page

No DVMRP, a árvore otimizada também não é predefinida. Nenhum roteador conhece de antemão qual é a árvore de caminhos mais curtos. Assim, a árvore é construída gradualmente. Quando um roteador recebe um pacote, ele o encaminha através de suas portas, *baseado no endereço de origem*, e contribui para a formação da árvore. O restante da árvore é montado pelos outros roteadores em *downstream*. A idéia deste protocolo era:

1. Evitar a formação de *loops*.
2. Evitar a duplicação de pacotes, isto é, nenhuma rede recebe mais de uma cópia. Além disso, o caminho percorrido pela cópia é o caminho mais curto até o destino.
3. Assinar membros dinamicamente.

RPF

A idéia original do DVMRP era utilizar o **Reverse Path Forwarding (RPF)**. No RPF, um roteador encaminha a cópia que percorreu a distância mais curta da origem até o roteador. Para determinar se o pacote percorreu de fato o caminho mais curto, o RPF usa a tabela de roteamento do RIP. Ele finge que precisa enviar um pacote para a origem e determina se a porta dada na tabela de roteamento é a mesma por onde o pacote chegou. A Figura 21.34 ilustra o conceito do RPF.

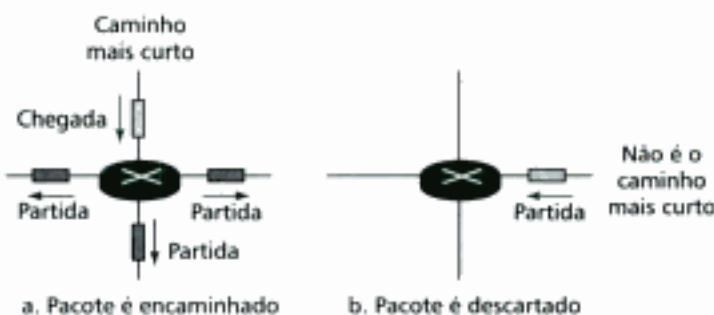


Figura 21.34 Encaminhando na direção reversa (RPF).

No RPF, o roteador encaminha somente os pacotes que percorreram o caminho mais curto da origem até o roteador; todas as cópias são descartadas. O RPF evita a formação de *loops* na rede.

RPB

O RPF garante a entrega de uma única cópia do pacote *multicast* em cada rede, sem a formação de *loops*. Contudo, o RPF não garante que cada rede recebe somente uma cópia. De fato, uma rede pode receber duas ou mais cópias do mesmo pacote. A razão disso é que o encaminhamento não é baseado em um endereço de destino (um endereço de grupo), mas baseia-se no endereço de origem. Para eliminar a duplicação devemos definir somente um roteador pai para cada rede. Uma rede pode receber um pacote *multicast* de uma origem particular somente através do *roteador designado pai*.

Nesse caso, a política fica clara. Para cada origem, o roteador envia um pacote através das portas designadas que passam pelo roteador pai. Esta política é denominada **Reverse Path Broadcasting (RPB)**. O RPB assegura que o pacote alcance as redes e cada rede recebe somente uma cópia do pacote. Observe a Figura 21.25 e perceba as diferenças entre RPF e RPB.

Você pode estar se perguntando: como o pai é escolhido? O roteador designado pai pode ser escolhido usando muitas estratégias diferentes; a mais comum delas é eleger o roteador que tiver o menor caminho até a origem como o roteador designado pai.

RPB cria uma árvore de broadcast de caminhos mais curta entre a origem e cada destino. Ele garante de fato que cada destino recebe uma, e somente uma, cópia do pacote.

Hidden page

Árvores de Custo Mínimo

Esta abordagem usa o fato do roteamento *link state (unicast)* ser montado a partir de uma base de dados comum e, baseado nela, cada roteador conhece toda a topologia da rede. Além disso, os roteadores podem utilizar o algoritmo de Dijkstra para criar a árvore de custo mínimo que possui um roteador designado como raiz e os demais roteadores da rede como nós da árvore. Entretanto, as árvores de custo mínimo geradas pelo algoritmo de Dijkstra no MOSPF são diferentes para cada roteador, ao contrário da base de dados e da topologia, que são as mesmas para cada roteador.

A árvore desejada no roteamento *multicasting* é um pouco diferente daquela utilizada no roteamento *unicast* de custo mínimo. No roteamento *multicast* precisamos de uma árvore para cada par fonte–grupo, onde a raiz deve estar localizada na fonte (a origem). A solução não é muito difícil porque a base de dados ainda é a mesma. Podemos exigir que cada roteador use o algoritmo de Dijkstra e crie uma árvore com a fonte na raiz. Neste caso, cada roteador cria exatamente a mesma árvore, sendo ele mesmo um nó na árvore. A Figura 21.37 mostra a diferença entre as árvores.

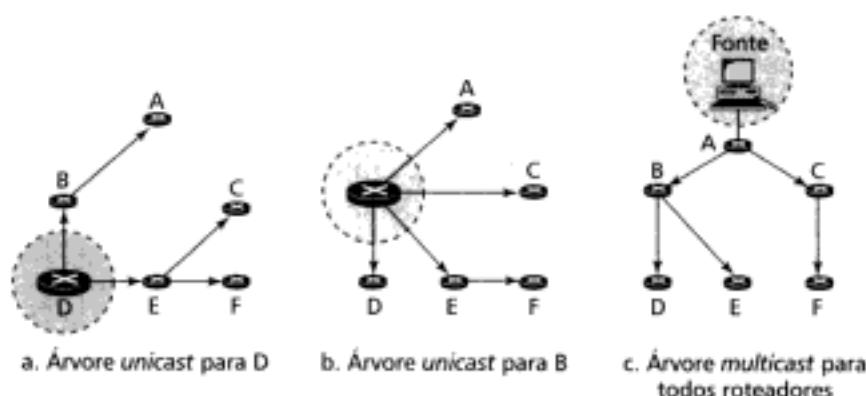


Figura 21.37 Árvores *unicast* e *multicast*.

Embora uma árvore feita dessa forma pareça uma solução perfeita, existem ainda alguns problemas.

1. A árvore baseada no algoritmo de Dijkstra usa endereços *unicast* (que são únicos para cada *host*); a árvore que estamos buscando requer um grupo de endereços que não sejam únicos (mais de um *host* pode pertencer a um grupo e um *host* pode pertencer a muitos grupos).
2. A assinatura pode ser modificada freqüentemente. Um *host* pode pertencer a um grupo num determinado momento e depois não.
3. O algoritmo de Dijkstra é muito complexo. Usar o algoritmo para cada pacote *multicast* fica muito caro do ponto de vista de tempo de processamento.

Para resolver o primeiro problema podemos adicionar um **pacote de *link state* atualizado** que associa o endereço *unicast* de um *host* com o endereço de grupo. Assim, os roteadores recebem *LSAs sobre um grupo de membros*. Podemos ainda incluir na árvore somente os *hosts* (usando os endereços *unicast* deles) pertencentes a um grupo particular. Além disso, construímos uma árvore contendo todos os *hosts* pertencendo a um grupo, mas nós usamos apenas endereços *unicast* dos *hosts* nos cálculos.

Os novos pacotes de *link state* também resolvem o segundo problema se eles forem enviados sempre que acontecerem mudanças na composição dos membros que estiverem assinando grupos.

O terceiro problema é resolvido fazendo um roteador calcular as árvores de custo mínimo sob demanda (quando ele receber o primeiro pacote *multicast*). Além do mais, a árvore pode ser salva na memória cache para uso futuro do mesmo par fonte–grupo. O MOSPF é um protocolo

que *reage rapidamente*, a primeira vez que o roteador MOSPF vê um pacote com um certo endereço de origem e de grupo, o roteador calcula a árvore de menor caminho.

CBT

O protocolo **CBT (Core-Based Tree)** é um protocolo *multicast* de árvore compartilhada que usa um núcleo como a raiz da árvore. O sistema autônomo (AS) é dividido em regiões e um núcleo (roteador central ou roteador *rendezvous*) é escolhido em cada região. O procedimento para escolha do roteador *rendezvous* é complexo e, por isso, o deixamos fora do escopo deste livro.

Formação da Árvore

Após o roteador *rendezvous* ser escolhido, os outros roteadores são informados do endereço *unicast* desse roteador. Assim, cada roteador envia uma mensagem *unicast* para entrar (*join*) mostrando que o roteador quer unir-se ao grupo. Esta mensagem é enviada através de todos os roteadores que estiverem localizados entre o transmissor e o roteador *rendezvous*. Os roteadores intermediários extraem a informação necessária da mensagem, tal como o endereço *unicast* da origem e a porta através da qual o pacote foi recebido, e as encaminham até o próximo roteador do percurso. A árvore é formada quando o roteador *rendezvous* receber todas as mensagens de *join* dos membros do grupo.

Se um roteador quiser deixar (*leave*) o grupo, basta enviar uma mensagem ao roteador *upstream* (logo acima). O roteador *upstream* remove o *link* para esse roteador da árvore e encaminha a mensagem ao roteador *upstream* imediatamente acima e assim por diante. A Figura 21.38 ilustra a árvore *multicast* compartilhada com o roteador *rendezvous*.

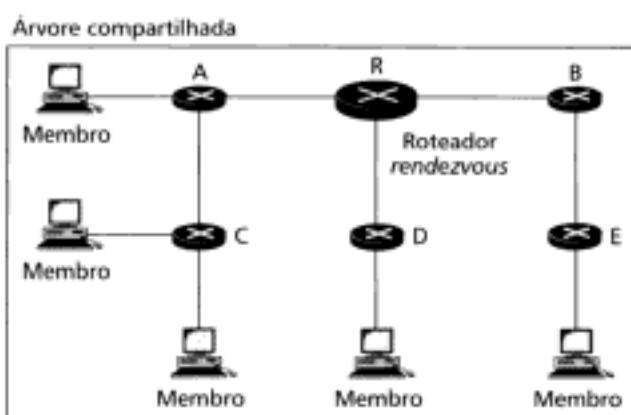


Figura 21.38 Árvore *multicast* compartilhada com roteador *rendezvous*.

O leitor deve ter observado duas diferenças essenciais entre o DVMRP e o MOSPF. Primeiro, a árvore dos dois primeiros é construída a partir da raiz, já a árvore CBT é formada de folhas. Segundo, no DVMRP a árvore é construída primeiramente (*broadcasting*) e então sofre um processo de poda (*pruned*). No CBT, não existe uma árvore de início, o processo de *joining* (*grafting*) constrói a árvore.

Enviando Pacotes Multicast

Após a formação da árvore, qualquer fonte (pertencente ao grupo ou não) pode enviar pacotes *multicast* a todos os membros do grupo. Ela envia simplesmente o pacote ao roteador *rendezvous*, usando o endereço *unicast* desse roteador. O roteador *rendezvous* distribui o pacote para todos os membros do grupo. Note que o *host* fonte pode ser qualquer um dos *hosts* dentro ou fora da árvore compartilhada. Na figura mostramos um *host* localizado fora da árvore compartilhada.

Em síntese, o CBT é uma árvore *multicast* compartilhada, um protocolo baseado em um ponto central, empregando uma árvore por grupo. Um dos roteadores na árvore é eleito o núcleo. Um pacote é enviado da fonte aos membros do grupo seguindo este procedimento:

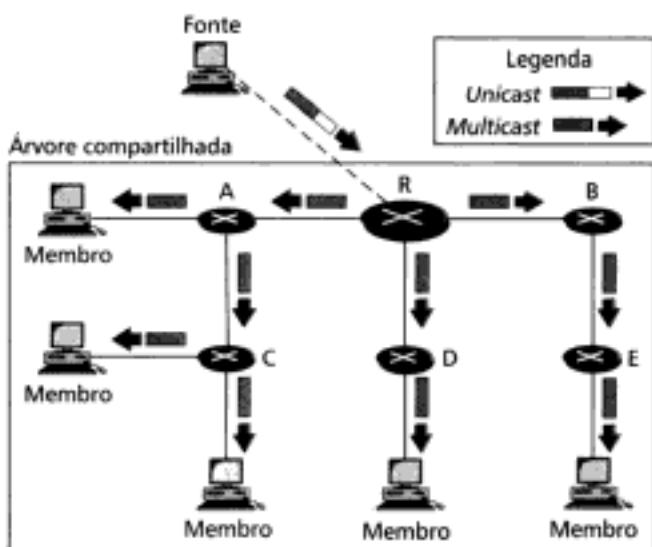


Figura 21.39 Enviando um pacote *multicast* para o roteador *rendezvous*.

1. A fonte, a qual pode ou não ser parte da árvore, encapsula o pacote *multicast* dentro de um pacote *unicast* com o endereço *unicast* de destino do núcleo e envia ao núcleo. Esta parte da entrega é realizada através do endereço *unicast*, o único receptor é o roteador núcleo (*core*).
2. O núcleo desencapsula o pacote *unicast* e o encaminha a todas as portas "interessadas", as quais são parte da árvore e não podem ser podadas pelo IGMP.
3. Cada roteador que recebe um pacote *multicast*, encaminha-o a todas as portas interessadas.

No CBT, a fonte envia um pacote *multicast* para o roteador *core* (núcleo). O roteador *core* desencapsula o pacote e o encaminha a todos os *hosts* interessados.

PIM

O protocolo PIM (Protocol Independent Multicast) é o nome dado a dois protocolos de roteamento independentes: **Protocol Independent Multicast, Dense Mode (PIM-DM)** e o **Protocol Independent Multicast, Sparse Mode (PIM-SM)**. À primeira vista ambos protocolos são parecidos, mas a semelhança termina aqui. Daremos uma breve descrição de cada um deles.

PIM-DM

O protocolo PIM-DM é independente de roteamento *unicast*, sendo bom para redes densas (muitos membros). Neste ambiente, o uso de um protocolo que produz pacotes *broadcast* é justificável porque quase todos os roteadores estão envolvidos no processo.

O protocolo PIM-DM é um protocolo de roteamento baseado na fonte que usa o RPF e as estratégias de *pruning/grafiting* para *multicasting*. A operação desse protocolo é semelhante ao DVMRP. Entretanto, diferentemente do DVMRP não depende de um protocolo de *unicast* específico. Ele assume que o AS está utilizando um protocolo de *unicast* e cada roteador possui uma tabela capaz de indicar a porta de saída para um caminho ótimo. O protocolo de *unicast* utilizado no AS pode ser um vetor de distâncias como o RIP ou um *link state* como o OSPF.

PIM-DM usa estratégias RPF, *pruning* e *grafting* para controlar o *multicasting*. Entretanto, ele é independente do protocolo *unicast* subjacente.

Hidden page

Hidden page

Hidden page

Hidden page

Hidden page

Hidden page

64. Quando um roteador *multicast* não está diretamente conectado a outro roteador *multicast*, um _____ pode ser formado para conectar os dois.

- Túnel físico
- Túnel lógico
- Núcleo lógico
- Spanning tree*

Exercícios

65. Em que se baseia a classificação para os quatro tipos de *links* definidos pelo OSPF?
66. Contraste e compare os roteamentos baseados no vetor de distâncias e *link state*.
67. Desenhe um fluxograma das etapas envolvidas quando um roteador recebe uma mensagem do vetor de distâncias de um vizinho.
68. Por que as mensagens OSPF propagam mais rapidamente que as mensagens RIP?
69. Um roteador possui a seguinte tabela de roteamento:

Net1	4	B
Net2	2	C
Net3	1	F
Net4	5	G

Como ficariam os itens da tabela se o roteador recebesse a seguinte mensagem RIP do roteador C?

Net1	2
Net2	1
Net3	3
Net4	7

70. Construa um Autonomous System (AS) contendo as seguintes especificações:
- Composto de oito rede (N1 a N8)
 - As redes são interligadas por oito roteadores
 - N1, N2, N3, N4 e N5 são redes Ethernet
 - N6 é uma rede *Token Ring*
 - N7 e N8 são redes ponto a ponto
 - R1 interliga N1 e N2
 - R2 interliga N1 e N7
 - R3 interliga N2 e N8
 - R4 interliga N7 e N6
 - R5 interliga N6 e N3
 - R6 interliga N6 e N4
 - R7 interliga N6 e N5
 - R8 interliga N8 e N5
71. Desenhe a representação gráfica do AS do Exercício 70 na visão do OSPF.
72. Que rede no Exercício 70 é uma rede transitória? Qual é a rede *stub*?

73. Por que não é necessário que uma mensagem IGMP viaje para fora da própria rede?
74. Uma lista de um roteador *multicast* possui quatro grupos (W, X, Y e Z). Há três hosts em cada LAN. O host A possui três membros leais (processos) assinando o grupo W e um membro leal assinando o grupo X. O host B possui dois membros leais assinando o grupo W e um membro leal assinando o grupo Y. O host C não tem processos assinando grupos. Apresente as mensagens IGMP envolvidas no monitoramento da rede.
75. Se um roteador possui 20 entradas numa tabela de grupos, ele deve enviar 20 mensagens de consulta (20 *queries*) ou apenas 1?
76. Se um host deseja continuar assinando cinco grupos, ele deve enviar 20 relatórios de assinatura ou apenas 1?
77. Um roteador com endereço IP 202.45.33.21 e endereço físico de rede Ethernet (MAC) 23:4A:45:12:EC:D2 transmite uma mensagem IGMP de consulta geral. Apresente todos os campos da mensagem.
78. Um host com endereço IP 124.15.13.1 e endereço físico de rede Ethernet 4A:22:45:12:E1:E2 envia um relatório IGMP de assinaturas sobre o grupo de ID 228.45.23.11. Apresente todos os campos da mensagem.
79. Um roteador numa rede Ethernet recebeu um pacote IP de *multicast* com ID de grupo 226.17.18.4. Quando o host verifica a tabela de grupos *multicast*, ele encontra este endereço. Mostre como o roteador transmite este pacote aos dispositivos receptores encapsulando o pacote IP em um *frame* Ethernet. Mostre todos os campos do *frame* Ethernet. O endereço IP da interface de saída do roteador é 185.23.5.6 e o endereço físico é 4A:22:45:12:E1:E2. Este roteador necessita dos serviços do protocolo ARP?
80. Um host cujo endereço IP é 114.45.7.9 recebe uma consulta IGMP. Quando o host verifica a tabela de grupos dele, ele

- não encontra as entradas. O que o *host* faz nessa situação? Ele deve enviar alguma mensagem? Caso a resposta seja sim, apresente os campos do pacote.
81. Um *host* cujo endereço IP é 222.5.7.19 recebe uma consulta IGMP. Quando o *host* verifica a tabela de grupos dele, ele encontra as seguintes entradas na tabela: 227.4.3.7 e 229.45.6.23. O que o *host* faz nessa situação? Ele deve enviar alguma mensagem? Caso a resposta seja sim, de que tipo e quantas mensagens ele precisa enviar? Apresente os campos.
82. Um *host* cujo endereço IP é 186.4.77.9 recebe uma solicitação de um processo de entrada (*join*) a um grupo cuja ID é 230.44.101.34. Quando o *host* verifica a tabela de grupos dele, descobre que não há nenhuma entrada relativa a essa ID. O que o *host* faz nessa situação? Ele deve enviar alguma mensagem? Caso a resposta seja sim, apresente os campos do pacote.
83. Um roteador cujo endereço IP é 184.4.7.9 recebe um relatório de um *host* que deseja entrar (*join*) a um grupo cuja ID é 232.54.10.34. Quando o roteador verifica a tabela de grupos dele, descobre que não há nenhuma entrada relativa a essa ID. O que o roteador faz nessa situação? Ele deve enviar alguma mensagem? Caso a resposta seja sim, apresente os campos do pacote.
84. Um roteador faz um consulta e recebe somente três relatórios sobre IDs de grupo: 225.4.6.7, 225.332.56.8 e 226.34.12.9. Quando o roteador olha a tabela de roteamento dele, encontra as seguintes entradas: 225.4.6.7, 225.11.6.8, 226.34.12.9, 226.23.22.67 e 229.12.4.89. O que o roteador faz nessa situação?
85. Um roteador usando protocolo DVMRP recebe da porta 2 um pacote cujo endereço IP de origem é 10.14.17.2. Supondo que o roteador encaminhe o pacote, quais são os itens da entrada, relacionados a este endereço IP, na tabela de roteamento *unicast*?
86. O roteador A envia pacote RIP *unicast* de atualização ao roteador B baseado na rede 134.23.0.0/16 a 7 saltos de distância. A rede B envia um pacote de atualização ao roteador A baseado na rede 13.23.0.0/16 a 4 saltos de distância. Se estes dois roteadores estão conectados no mesmo AS, qual deles é o roteador designado pai? Por quê?
87. O protocolo RPF pode de fato desempenhar o mecanismo *spanning tree*? Explique.
88. O protocolo RPB pode de fato desempenhar o mecanismo *spanning tree*? Explique.
89. O protocolo RPM pode de fato desempenhar o mecanismo *spanning tree*? Explique.

PARTE V

CAMADA DE TRANSPORTE

A camada de transporte é o núcleo do modelo da Internet. Os protocolos dessa camada supervisionam o fluxo de dados entre processos finais, isto é, entre um programa rodando em um computador e outro programa rodando noutro computador. Mais importante que isso, tais protocolos agem como uma conexão entre os protocolos da camada de aplicação e os serviços oferecidos pelas camadas mais baixas (rede, enlace e física). Os programas da camada de aplicação interagem uns com os outros usando os serviços fornecidos pela camada de transporte, sem mesmo ter que tomar conhecimento da existência das camadas inferiores. Em outras palavras, as redes físicas são transparentes para os processos da camada de aplicação, não sendo nem mesmo dependentes do tipo de rede física utilizada. Para os processos da camada de aplicação, as redes físicas são simplesmente uma nuvem homogênea que, de alguma forma, tomam os dados e os entregam, em segurança, ao destino final.

A Figura 1 mostra a posição da camada de transporte no modelo de cinco camadas da Internet. A camada de transporte é a quarta camada no modelo. Acima dela está situada a camada de aplicação e abaixo está situada a camada de rede. Isto significa que a camada de transporte recebe os serviços da camada de rede e oferece serviços à camada de aplicação.

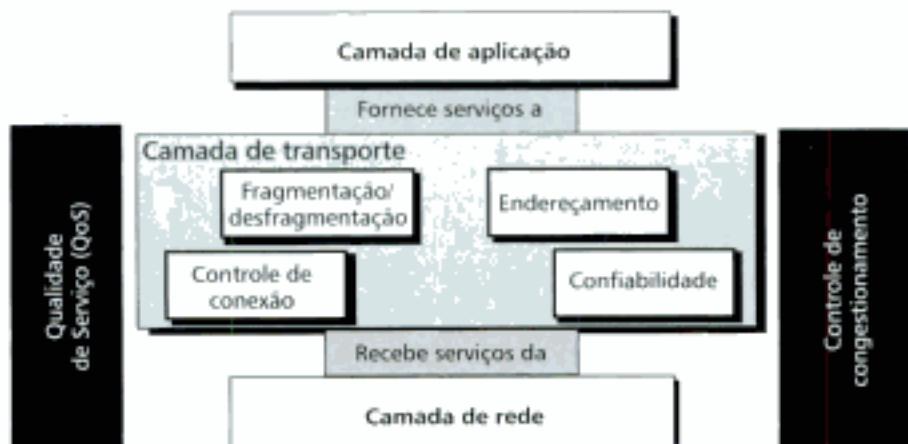


Figura 1 Posição da camada de transporte.

Serviços

O fluxo de informação entre processos finais é realizado através de um conjunto de funcionalidades ligadas à camada de transporte. Os serviços mais importantes dessa camada são encapsulamento, controle da conexão, endereçamento e confiabilidade, como ilustra a Figura 2.

Estes serviços serão discutidos em detalhes no Capítulo 22. A seguir faremos um resumo do que será estudado nos Capítulos 22 e 23.



Figura 2 Serviços da camada de transporte.

Encapsulamento

A camada de transporte cuida dos processos de montagem dos pacotes recebidos da camada de aplicação. O processo de encapsulamento divide grandes mensagens em segmentos menores. Tais segmentos são devidamente encapsulados no campo de dados (como *payload*) do pacote da camada de transporte, sendo devidamente identificados através da informação inserida no cabeçalho.

Dividindo Mensagens Grandes Uma mensagem gerada na camada de aplicação pode variar em tamanho. Por exemplo, um cliente SMTP (protocolo de *e-mail*) pode enviar uma mensagem curta (texto em poucas linhas) ou uma mensagem longa constituída de muitos documentos anexados, talvez até aplicações de multimídia. Uma mensagem longa pode superar o tamanho do campo de *payload* dos protocolos da camada inferior. Por exemplo, alguns tipos de camada de rede conseguem controlar somente pacotes com um pouco mais de mil caracteres. Daí, as mensagens longas oriundas da camada de aplicação devem ser divididas em segmentos menores, cada qual encapsulado em um pacote separado. De certa forma, isto é muito parecido com o que acontece quando desejamos enviar uma carta muito extensa (com muitas páginas) e dispomos apenas de envelopes de tamanho padrão. Precisamos separar as páginas em envelopes individuais e endereçar todas as cartas ao mesmo destinatário.

Adicionando um Cabeçalho A camada de transporte encapsula o pacote e adiciona informações de cabeçalho. Isso acontece até mesmo se a mensagem oriunda da camada de aplicação é pequena o suficiente para ser controlada diretamente pela camada de rede. O cabeçalho permite que a camada de transporte realize outras funções.

Controle da Conexão

Os protocolos da camada de transporte são divididos em dois tipos: orientados à conexão e sem conexão.

Serviço Orientado à Conexão Um protocolo de transporte orientado à conexão estabelece inicialmente um circuito virtual entre as aplicações dos usuários finais. Perceba que a conexão é virtual, isto é, para a camada de aplicação existe apenas um único caminho entre as aplicações finais. Entretanto, na realidade os pacotes podem viajar através de muitos caminhos diferentes. É comum dizermos que foi aberta uma sessão entre usuários finais. A sessão permanece intacta até que seja solicitada uma desconexão por uma das partes. Neste meio tempo, as partes envolvidas podem transmitir pacotes à vontade, viajando um após o outro através do circuito virtual (do ponto de vista da camada de aplicação). É claro que os pacotes podem viajar fora de ordem, mas a camada de

transporte toma medidas para assegurar que isto seja transparente para a aplicação. Os pacotes são numerados em seqüência e a comunicação pode acontecer em modo *full-duplex*.

Serviço sem Conexão Um protocolo de transporte sem conexão trata cada pacote independentemente, sem qualquer conexão entre eles.

Endereçamento

Imagine que um cliente HTTP (um *browser* de Internet), instalado em um computador local, necessite enviar uma solicitação a um servidor HTTP remoto. Primeiro, o cliente necessita endereçar unicamente o computador remoto. Este esquema de endereçamento é implementado na camada de rede, conforme vimos no Capítulo 19. No momento, vamos assumir que o cliente conhece o endereço do computador remoto.

Há, contudo, outro problema a ser resolvido. O computador remoto possivelmente está rodando muitas aplicações ao mesmo tempo, tal como HTTP, SMTP e TELNET. Quando a solicitação chega ao computador remoto, ele deve repassá-la apenas ao programa servidor HTTP. Em outras palavras, o pacote transportando a solicitação deve especificar que tipo de programa servidor deve receber a solicitação.

O pacote de solicitação deve também especificar o programa cliente que enviou o pacote. O servidor faz uso dessa informação quando responde à solicitação. A razão principal é que o cliente também pode estar rodando muitas aplicações cliente. As pessoas normalmente abrem muitos programas cliente ao mesmo tempo, idênticos ou diferentes.

Confiabilidade

A camada de transporte oferece confiabilidade à aplicação que utiliza os seus serviços. Confiabilidade, como vimos na camada de enlace, envolve controle de fluxo e controle de erros.

Controle de Fluxo A camada de transporte, tal como a camada de enlace, oferece controle de fluxo. Entretanto, o controle de fluxo desta camada é realizado entre processos finais (*process-to-process*), em vez de usar um único *link*.

Controle de Erro A camada de transporte, tal como a camada de enlace, oferece controle de erros. Entretanto, o controle de erros desta camada é realizado entre processos finais. A camada de transporte do dispositivo transmissor certifica-se que toda a mensagem chega à camada de transporte do dispositivo receptor livre de erros (danos, perda ou duplicação). A correção de erros é realizada usualmente através de retransmissão.

Controle de Congestionamento e QoS

Embora o congestionamento na rede possa acontecer nas camadas de enlace, rede ou transporte, o efeito do congestionamento normalmente é sentido na camada de transporte porque esta é a camada que oferece serviços diretos à camada de aplicação. Ainda não tratamos o controle de congestionamento em nenhuma parte do texto. Chegou a hora de lidarmos com este aspecto.

Qualidade de Serviço (*QoS*) é outro assunto que postergamos nos capítulos passados. Embora a *QoS* possa ser implementada em outras camadas, os efeitos notórios são observados na camada de transporte.

Trataremos o controle de congestionamento e a qualidade de serviços no Capítulo 23.

Capítulos

Incluímos dois capítulos nessa parte do livro: Capítulo 22, onde serão abordados os conceitos e os serviços da camada de transporte e serão explicados os dois protocolos da camada de transporte do modelo da Internet: TCP e UDP. O Capítulo 23 aborda duas questões fundamentais associadas à camada de transporte: controle de congestionamento e qualidade de serviço.

Protocolos da Camada de Transporte: TCP e UDP

Iniciaremos o capítulo justificando a necessidade da camada de transporte: comunicação envolvendo processos finais. Discutiremos as questões que resultam deste tipo de comunicação e analisaremos alguns métodos para controlá-la.

O modelo da Internet possui dois protocolos da camada de transporte: TCP e UDP. Iniciaremos estudando as características do UDP, porque é o mais simples dos dois. Em seguida, veremos como utilizar este protocolo simples que carece de muitas características do TCP.

Finalizaremos o capítulo abordando o TCP. O TCP é um protocolo da camada de transporte bastante complexo. Veremos como os conceitos previamente definidos são aplicados ao TCP. Deixaremos as discussões sobre alguns serviços disponíveis no nível de transporte, como o controle de congestionamento e qualidade de serviço (QoS), para o Capítulo 23. Estes dois tópicos aplicam-se também às camadas de enlace e camada de rede.

22.1 COMUNICAÇÃO ENTRE PROCESSOS FINAIS

A camada de enlace é responsável pelos processos de entrega de *frames* entre dois nós vizinhos conectados em um *link*. Isto é denominado *comunicação nó a nó (node-to-node)*. Vimos que a camada de rede faz os melhores esforços para entregar pacotes entre dois *hosts*. Isto é denominado *comunicação entre hosts (host-to-host)*. A comunicação na Internet não é definida como a troca de informações entre dois nós ou entre dois *hosts*. Uma comunicação real acontece entre dois processos finais (programas aplicativos). Assim, precisamos discutir a **comunicação entre processos finais (process-to-process)**. Entretanto, em qualquer momento, muitos processos podem estar rodando tanto no *host* de origem quanto no *host* de destino. Para completar a entrega de pacotes, precisamos de um mecanismo para entregar dados de um processo rodando na origem para outro processo rodando no *host* de destino.

A camada de transporte cuida da comunicação entre tais processos finais; da entrega de um pacote, parte de uma mensagem, de um processo até o outro processo. Veremos mais tarde que dois processos se comunicam numa abordagem cliente-servidor. A Figura 22.1 ilustra os três tipos de comunicação e os domínios onde são encontrados.

A camada de transporte é responsável pela comunicação envolvendo processos finais.

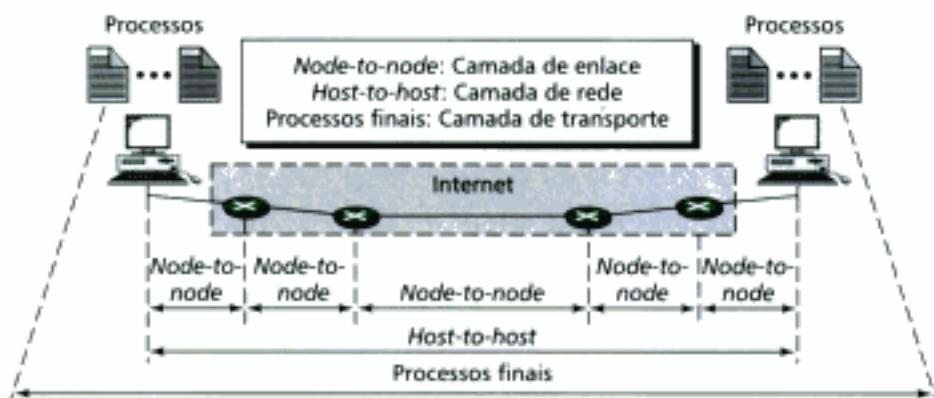


Figura 22.1 Tipos de comunicação.

Paradigma Cliente-Servidor

Embora existam muitos modos de realizar uma comunicação entre processos finais, talvez o mais comum deles é através do **paradigma cliente-servidor**. Um processo em que um *host* local, denominado **cliente**, precisa dos serviços de outro processo localizado em um *host* remoto, denominado **servidor**.

Ambos processos (cliente e servidor) possuem o mesmo nome. Por exemplo, para obter o dia e a hora de uma máquina remota necessitamos de um processo cliente, denominado *Daytime*, rodando no *host* local e um processo servidor, *Daytime* também, rodando no *host* remoto.

Hoje, os sistemas operacionais suportam tanto o ambiente multiusuário quanto o ambiente multiprogramação. Um computador remoto pode rodar diversas aplicações servidoras, tanto quanto os computadores locais podem rodar um ou mais programas cliente ao mesmo tempo. Para a comunicação devemos definir o seguinte:

1. *Host* local
2. Processo local
3. *Host* remoto
4. Processo remoto

Mecanismo de Endereçamento

Sempre que necessitarmos entregar dados a um destino específico, dentre muitos, devemos utilizar algum esquema de endereçamento. Vimos que na camada de enlace o endereço MAC se encarrega de endereçar um nó dentre vários, se a conexão não é ponto a ponto. Um *frame* no nível de enlace necessita do endereço MAC de destino para que o pacote seja entregue e um endereço MAC de origem para que a origem possa ser respondida.

Na camada de rede, precisamos de um endereço IP para discriminar um *host* entre muitos outros (na Internet seria discriminar um entre milhões). Um datagrama da camada de rede precisa dos endereços IP de origem e de destino para que a origem estabeleça a comunicação e o destino possa respondê-la.

Na camada de transporte também existe um esquema de endereçamento, denominado **número de porta**, para discriminar entre os muitos processos que possivelmente estão rodando no *host* de destino. O número de porta de destino é necessário para identificar o endereço de entrega e o número de porta de origem para identificar o endereço de resposta.

Hidden page

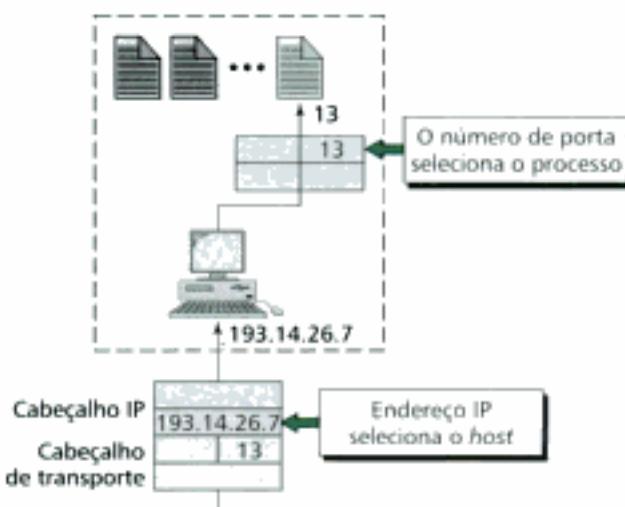


Figura 22.3 Endereços IP versus números de portas.



Figura 22.4 Faixas IANA.

Endereços de Socket

A comunicação envolvendo processos finais precisa de dois identificadores: endereço IP e número de porta. À combinação de um endereço IP e um número de porta damos o nome de **endereço de socket**. O endereço de *socket* do cliente define o processo cliente univocamente, assim como o endereço de *socket* do servidor também o define complementarmente (veja a Figura 22.5).

Um protocolo de transporte necessita de um par de endereços de *socket*: o endereço *socket* do cliente e o endereço *socket* do servidor. Estas quatro partes da informação são parte do cabeçalho do pacote IP e do cabeçalho do protocolo da camada de transporte. O cabeçalho IP contém os endereços IP (origem e destino). O cabeçalho TCP ou UDP contém os números de portas.

Multiplexação e Demultiplexação

O mecanismo de endereçamento permite que a camada de transporte de origem multiplexe os dados das aplicações finais ativas no cliente e a camada de transporte de destino demultiplexe os dados para as aplicações finais ativas no servidor, conforme ilustra a Figura 22.6.

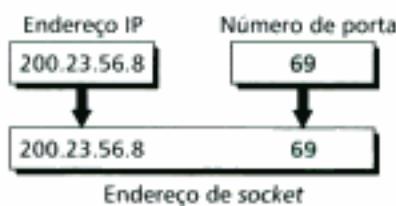


Figura 22.5 Endereço de socket.

Hidden page

Hidden page

Hidden page

Hidden page

Hidden page

Hidden page

Serviços TCP

Vamos explicar os serviços oferecidos pelo TCP aos processos da camada de aplicação.

Fluxo do Serviço de Entrega

O TCP, diferentemente do UDP, é um protocolo que orienta o fluxo de comunicação. Um processo da camada de aplicação envia um segmento de dados ao UDP na camada de transporte para iniciar a entrega desses dados. O UDP adiciona um cabeçalho próprio ao conjunto, montando o datagrama do usuário, e o entrega ao protocolo IP para transmissão. O processo pode enviar diversos segmentos de dados ao UDP, mas o UDP trata cada segmento independentemente, sem procurar estabelecer conexão entre os segmentos.

Por outro lado, o TCP permite ao processo transmissor entregar dados como uma cadeia de *bytes* e ao processo receptor obter os dados como uma cadeia de *bytes*. O TCP cria um ambiente no qual os dois processos parecem estar conectados por um "tubo" imaginário por onde são transportados os dados através da Internet. Este ambiente imaginário aparece descrito na Figura 22.11. O processo transmissor produz um fluxo de dados (cadeia) e o processo receptor o consome.



Figura 22.11 Fluxo do serviço de entrega.

Buffers de Transmissão e Recepção

Visto que os dois processos (transmissão e recepção) podem não produzir e consumir dados à mesma velocidade, o TCP necessita do esquema de *bufferização*. O TCP requer dois *buffers*: um para transmissão e outro para a recepção (um em cada direção). Veremos adiante que tais *buffers* também são utilizados pelos mecanismos de controle de fluxo e erros. Um modo de implementar um *buffer* é usar um arranjo circular de 1-byte, tipo FIFO (*first-in, first-out*), como mostra a Figura 22.12.

Para simplificar mostramos dois *buffers* de 20 *bytes* cada. Normalmente, os *buffers* têm centenas ou milhares de *bytes*, dependendo da implementação. Na figura, ambos têm o mesmo tamanho, o que não é sempre o caso.

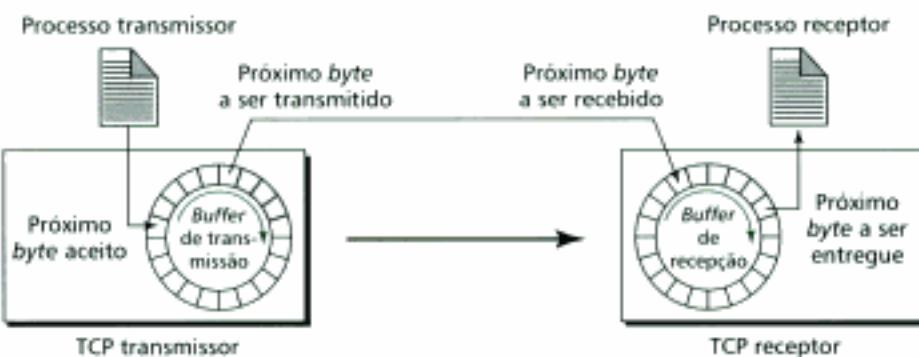


Figura 22.12 Buffers de transmissão/recepção.

Hidden page

1. O protocolo TCP que cuida dos processos de A solicita ao protocolo TCP de B, que cuida dos processos de B, aprovação para iniciar a transmissão.
2. O TCP de A e B trocam dados em ambas direções.
3. Quando ambos processos não tiverem dados a transmitir, os *buffers* tornam-se vazios. Os protocolos TCP nos dois lados reciclam os respectivos *buffers*.

Observe que isto é uma conexão virtual, não uma conexão física. O segmento TCP é encapsulado em um datagrama IP e, por isso, pode chegar fora de ordem, perder-se, ser corrompido e, então, precisar ser retransmitido. Cada segmento pode seguir um caminho diferente até o destino, porque não existe uma conexão física dedicada entre as duas pontas. Entretanto, o TCP cria o ambiente orientado à conexão, aceitando para si a responsabilidade de entregar os *bytes* de dados no outro lado. É como se uma ponte fosse criada interligando múltiplas ilhas com o tráfego indo de uma ilha para outra através de uma única conexão.

Serviço Confiável

O TCP é um protocolo de transporte confiável. Ele utiliza um mecanismo de confirmação (ACK) para verificar a integridade dos dados. Veremos esta característica na seção sobre controle de erros.

Numeração de Bytes

Embora o protocolo TCP mantenha os registros dos segmentos transmitidos e/ou recebidos, não há um campo específico para o número do segmento. Ao invés disso, há dois campos denominados **número de seqüência** e **número de confirmação**. Estes dois campos referem-se ao número do *byte* e não ao número do segmento propriamente dito.

Números de Bytes

O protocolo TCP numera todos os *bytes* de dados que são transmitidos em uma conexão. A numeração é independente em cada direção. O TCP numera os *bytes* quando recebe dados de um processo e os armazena no *buffer* de transmissão. A numeração não começa necessariamente em 0. O valor inicial é estabelecido por um gerador de números aleatórios entre 0 e $2^{32} - 1$. Por exemplo, se o número aleatório escolhido for 1057 e a quantidade total de *bytes* é 6000 *bytes*, os *bytes* são numerados de 1057 a 7056. Veremos que o esquema de numeração de *bytes* é utilizado durante o controle de fluxo e de erros.

Os *bytes* de dados transferidos em cada conexão são numerados pelo TCP. A numeração inicial é escolhida a partir de um gerador de números aleatórios.

Número de Seqüência

Após os *bytes* terem sido numerados, o TCP atribui um número de seqüência para cada segmento transmitido. O número de seqüência do segmento é idêntico ao número do primeiro *byte* transportado nesse segmento.

Exemplo 1

Imagine uma conexão TCP transferindo 6000 *bytes*. O primeiro *byte* recebe a numeração 10010. Quais são os números de seqüência para cada segmento se os dados são transmitidos em cinco segmentos, os quatro primeiros segmentos transportando 1000 *bytes* e o último transportando 2000 *bytes*?

Solução

A seguir vemos os números de seqüência para cada segmento:

- Segmento 1 → Número de seqüência: 10.010 (Faixa: 10.010 a 11.009)
- Segmento 2 → Número de seqüência: 11.010 (Faixa: 10.010 a 12.009)
- Segmento 3 → Número de seqüência: 12.010 (Faixa: 12.010 a 13.009)
- Segmento 4 → Número de seqüência: 13.010 (Faixa: 13.010 a 14.009)
- Segmento 5 → Número de seqüência: 14.010 (Faixa: 14.010 a 16.009)

Hidden page

Hidden page

Conexão

Sabemos que o TCP é um protocolo orientado à conexão. Ele estabelece um caminho virtual entre a origem e o destino. Todos os segmentos pertencentes a uma mensagem são enviados através desse caminho virtual. Usar um único caminho virtual para toda mensagem facilita o processo de confirmação, além de facilitar a retransmissão de frames danificados ou perdidos (caso ocorra). No TCP, a transmissão orientada à conexão requer dois procedimentos: **estabelecimento** e **término da conexão**.

Estabelecimento da Conexão

Vimos que o TCP transmite dados em modo *full-duplex*. Quando os protocolos TCP de duas máquinas são conectados, eles são capazes de trocar segmentos simultaneamente. Isto implica que cada parte deve iniciar a comunicação e obter aprovação da outra antes que os dados comecem a ser transferidos. Conforme discutimos antes são necessárias quatro etapas para o estabelecimento da conexão. Entretanto, a segunda e terceira etapas podem ser combinadas para criar uma conexão em três etapas denominado **handshake triplo (three-way handshake)**, conforme ilustra a Figura 22.16.

As etapas do processo são as seguintes:

1. Um cliente inicia uma conexão enviando um pacote que indica seu *número de seqüência inicial (ISN)*, com um determinado *bit* no cabeçalho predefinido, indicando um pedido de conexão. O segmento SYN inclui os números das portas de origem e destino. O número da porta de destino define claramente a aplicação para a qual os dados devem ser entregues.
2. O servidor envia o segundo segmento, um segmento SYN e ACK. Este segmento possui duplo propósito. Primeiro, confirma o recebimento do segmento inicial, usando o *flag* de ACK e o campo número de confirmação. Observe que o número de confirmação é o ISN do cliente acrescido de 1, visto que não foram enviados dados no segmento 1. O servidor deve definir também o tamanho da janela deslizante do cliente. Segundo, o segmento é utilizado como segmento de inicialização pelo servidor. Ele contém o ISN que numera os *bytes* indo do servidor ao cliente.
3. O cliente envia o terceiro segmento. Trata-se apenas de um segmento ACK. Esse segmento confirma a recepção do segundo segmento, usando o *flag* ACK e o campo número de confirmação. Observe que o número de confirmação é o ISN do servidor acrescido de 1, porque não foram enviados dados no segmento 2. O cliente deve definir o tamanho da janela deslizante do servidor. Dados podem ser transmitidos no terceiro pacote.

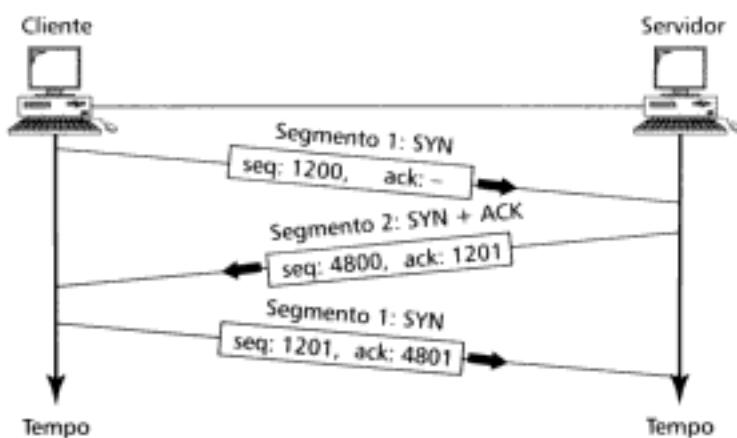


Figura 22.16 As três etapas de estabelecimento da conexão.

Término da Conexão

Qualquer uma das duas partes envolvidas no processo de troca de dados (cliente ou servidor) pode encerrar a conexão. Quando a conexão é finalizada em apenas uma direção, a outra parte pode continuar transmitindo dados na outra direção. Sendo assim, as quatro etapas são necessárias para encerrar a conexão em ambas direções, como mostra a Figura 22.17.

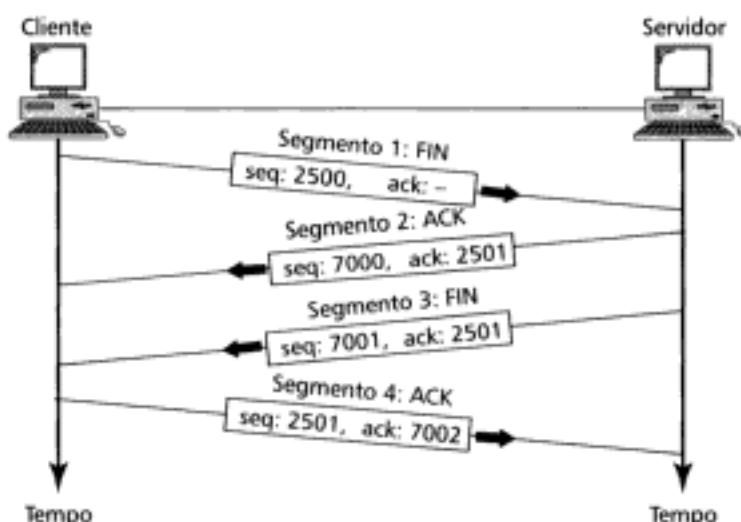


Figura 22.17 As quatro etapas para o término da conexão.

As quatro etapas envolvidas no processo de desconexão são as seguintes:

1. O cliente TCP transmite o primeiro segmento, um segmento FIN.
2. O servidor TCP transmite o segundo segmento, um segmento ACK, para confirmar o recebimento do segmento FIN do cliente. Observe que o número de confirmação é igual ao número de seqüência do segmento FIN mais 1.
3. O servidor TCP pode continuar transmitindo dados na direção servidor-cliente. Quando o servidor não tiver mais dados a enviar, ele transmite o segmento FIN.
4. O cliente TCP envia um quarto segmento, um segmento ACK, para confirmar o recebimento do segmento FIN do servidor TCP. Observe que o número ACK é igual ao número de seqüência recebido do segmento FIN mais 1.

Resetting da Conexão

O TCP pode solicitar ainda o *resetting* da conexão. *Resetting* aqui significa que a conexão atual é desligada sumariamente. Isto acontece em três situações:

1. O TCP do cliente solicitou uma conexão a uma porta não existente. O TCP do servidor pode enviar um segmento com o bit RST configurado para anular a solicitação.
2. O TCP do servidor pode querer abortar a conexão devido a uma situação anormal. Ele pode enviar um segmento RST para encerrar a conexão.
3. O TCP do cliente pode descobrir que o TCP do servidor tem ficado ocioso por muito tempo. Ele pode enviar um segmento RST para desligar a conexão.

Diagrama de Transição de Estados

Para manter em ordem todos os eventos diferentes que acontecem durante o estabelecimento ou término de uma conexão e a eventual transferência de dados, o software TCP é implementado como uma máquina de estados finitos. Uma **máquina de estados finitos** trabalha sempre com um

número limitado de estados. Em qualquer momento, a máquina está em um desses estados. Ela permanece em um estado até que aconteça um evento. O evento pode conduzir a máquina para um novo estado ou então pode fazer a máquina realizar algumas ações. Em outras palavras, o evento é uma entrada aplicada ao estado. Ela pode modificar o estado e também pode criar uma saída. A Tabela 22.4 mostra os estados do protocolo TCP.

TABELA 22.4 Estados do Protocolo TCP

Estado	Descrição
CLOSED	Sem conexão.
LISTEN	O servidor está aguardando uma chamada do cliente.
SYN-SENT	Uma solicitação de conexão foi enviada; esperando pelo ACK.
SYN-RCVD	Uma conexão solicitada foi recebida.
ESTABLISHED	A conexão foi estabelecida.
FIN-WAIT-1	A aplicação solicitou o encerramento da conexão.
FIN-WAIT-2	A outra parte aceitou o encerramento da conexão.
TIME-WAIT	Aguardando pelos segmentos retransmitidos para finalizar.
CLOSE-WAIT	O servidor está aguardando a aplicação para encerrar.
LAST-ACK	O servidor está aguardando pelo último ACK.

Para ilustrar o conceito usamos um **diagrama de transição de estados**. Os estados são todos mostrados no fluxograma. A transição de um estado para o outro é mostrada através das linhas de transição. Para cada linha de transição são feitos dois comentários, separados por uma barra (/). O primeiro comentário é a entrada que o TCP recebe. O segundo é a saída que o TCP gera. A Figura 22.18 mostra o diagrama de transição de estados cliente-servidor. As linhas tracejadas da figura representam os estados do servidor e as linhas cheias representam os estados do cliente. O diagrama real é mais complexo que o mostrado na figura.

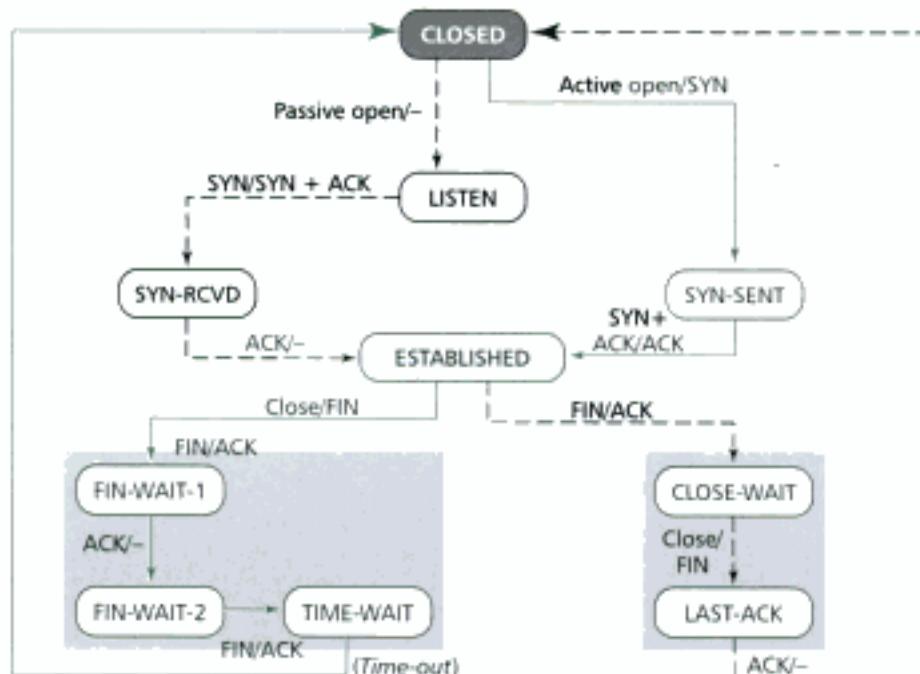


Figura 22.18 Diagrama de transição de estados.

Diagrama Cliente

O cliente possui os seguintes estados: CLOSED, SYN-SENT, ESTABLISHED, FIN-WAIT-1, FIN-WAIT-2 e TIME-WAIT.

- O cliente TCP inicia no estado CLOSED.
- Enquanto estiver neste estado, o cliente TCP pode receber uma solicitação *active open* da aplicação cliente. O cliente envia um segmento SYN ao servidor TCP e evolui para o estado SYN-SENT.
- Nesse estado, o cliente TCP espera receber um segmento SYN + ACK do protocolo TCP servidor. O cliente envia um segmento ACK ao servidor TCP e evolui para o estado ESTABLISHED. Nesse estado, ocorre a transferência de dados. O cliente permanece assim enquanto houver transmissão e recepção de dados.
- O cliente TCP espera receber uma solicitação de encerramento da aplicação cliente. O cliente envia um segmento FIN para o servidor TCP e evolui para o estado FIN-WAIT-1.
- Nesse estado, o cliente TCP espera receber um ACK do servidor TCP. Quando o ACK é recebido, o cliente evolui para o estado FIN-WAIT-2. Não é enviado nada, ocorre apenas o encerramento da conexão em uma direção.
- O cliente permanece nesse estado aguardando o servidor fechar a conexão na outra direção. Se o cliente receber um segmento FIN do servidor, envia um segmento ACK e evolui para o estado TIME-WAIT.
- Quando o cliente evolui para esse estado, ele dispara automaticamente um relógio (*timer*). O valor deste *timer* é configurado para duas vezes o tempo de vida estimado de um segmento de tamanho máximo. O cliente permanece no estado TIME-WAIT antes de encerrar completamente a conexão. Isso possibilita que os pacotes duplicados, caso existam, cheguem ao destino para serem descartados. Após o *time-out* (interrupção), o cliente retorna ao estado CLOSED.

Diagrama Servidor

Embora o servidor possa estar em qualquer um dos 11 estados, geralmente, na operação normal, ele está em um dos seguintes estados: CLOSED, LISTEN, SYN-RVD, ESTABLISHED, CLOSED-WAIT e LAST-ACK.

- O servidor TCP inicia no estado CLOSED.
- Enquanto estiver nesse estado, o servidor TCP espera receber uma solicitação *passive open* da aplicação servidora. Se assim for, o servidor TCP evolui para o estado LISTEN.
- O servidor TCP espera receber um segmento SYN do cliente TCP. Ele envia um segmento SYN + ACK ao cliente TCP e evolui para o estado SYN-RCVD.
- O servidor TCP espera receber um segmento ACK do cliente TCP. Caso receba, ele evolui para o estado ESTABLISHED. Nesse estado ocorre a transferência de dados. O servidor permanece no estado ESTABLISHED enquanto estiver transmitindo e recebendo dados.
- O servidor TCP espera receber um segmento FIN do cliente, o que indica que o cliente deseja terminar a conexão. O servidor pode enviar um segmento ACK ao cliente e evoluir para o estado CLOSE-WAIT.
- No estado CLOSE-WAIT, o servidor espera até receber uma solicitação de término do programa servidor. Então, ele envia um segmento FIN ao cliente e evolui para o estado LAST-ACK.
- Nesse estado, o servidor espera receber o último segmento ACK. Daí, o servidor TCP retorna ao estado inicial CLOSED.

Controle de Fluxo

O **controle de fluxo** estabelece a quantidade de dados que um *host* de origem pode transmitir antes de receber uma confirmação (ACK) do *host* de destino. Em casos extremos, um protocolo de transporte poderia enviar um *byte* de dados e esperar pelo ACK antes de transmitir o próximo *byte*. Mas isso geraria um processo de comunicação extremamente lento. Se os dados tiverem que percorrer uma distância física muito grande, o *host* de origem ficaria muito tempo ocioso, esperando por um ACK.

No outro extremo, um protocolo de transporte pode transmitir todos os dados de uma única vez, sem se preocupar em receber ACKs. Isto pode aumentar a velocidade do processo de comunicação, mas pode sobrecarregar o receptor. Além disso, se alguma parte da massa de dados for perdida, duplicada, recebida fora de ordem ou corrompida, o *host* de origem não saberá até que todos os dados tenham sido verificados pelo *host* de destino.

O TCP apresenta uma solução entre esses dois extremos. Ele define uma janela que obriga o *buffer* de dados alimentado com os dados da aplicação transmissora. O TCP envia tantos dados quantos forem definidos pelo mecanismo baseado numa janela deslizante (móvel).

O Mecanismo Janela Deslizante

O janelamento é um mecanismo de controle de fluxo que exige que o dispositivo de origem receba uma confirmação do destino depois de transmitir uma determinada quantidade de dados. Neste método, ambos *hosts* usam uma janela para cada conexão. A janela ocupa uma porção do *buffer* contendo *bytes* que o *host* transmissor pode enviar antes de se preocupar em receber um ACK do receptor. O mecanismo é denominado **janela móvel** ou **janela deslizante** porque ocorre uma movimentação sobre o *buffer* de dados à medida que dados e confirmações são enviados e recebidos.

Uma janela deslizante é utilizada para tornar a transmissão mais eficiente, bem como controlar o fluxo de dados de forma que o *host* de destino não seja sobrecarregado. A janela deslizante do TCP é uma conexão orientada a *byte*.

A Figura 22.19 mostra o *buffer* transmissor da Figura 22.12. Entretanto, ao invés de representá-lo circularmente, mostramos um *buffer* plano para simplificar a explicação. Observe que, se as duas extremidades do *buffer* estiverem conectadas, implementamos novamente a ideia do *buffer* cíclico.

Na Figura 22.1, os *bytes* antes da posição 200 foram enviados e receberam confirmação. O transmissor pode reutilizar estas localizações. Os *bytes* entre as posições 200 e 202 foram enviados, mas estão aguardando confirmação. O transmissor deve mantê-los no *buffer* pois pode ser necessário retransmiti-los (em caso de perda ou extravio). Os *bytes* entre as posições 203 e 211 estão no *buffer* (foram produzidos pelo processo), mas ainda não foram transmitidos.

Vamos examinar a situação sem a implementação do mecanismo de janela móvel implementado. Nesse caso, o transmissor pode seguir enviando todos os *bytes* que estiverem no *buffer* (até

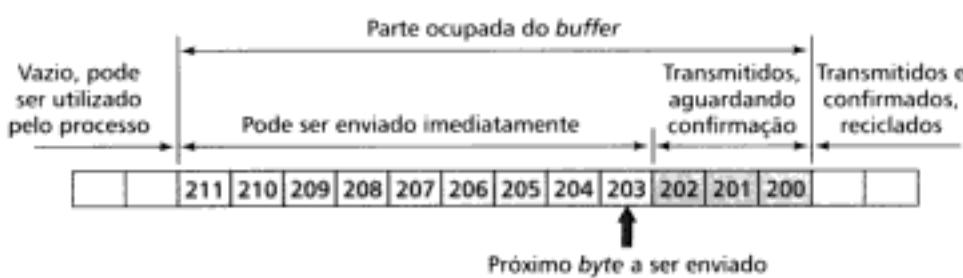


Figura 22.19 Buffer transmissor.

o 211) sem considerar o *status* do receptor. O *buffer* receptor possui tamanho limitado e poderia ser completamente esgotado pelo processo que está recebendo os dados mas não os está consumindo a uma velocidade suficiente para manter parte do *buffer* vazio. Os *bytes* excedentes seriam sumariamente descartados pelo receptor, isto é, sem solicitar retransmissão. Isso justifica a necessidade do transmissor ajustar-se à capacidade de recepção do receptor.

Janela de Recepção

A Figura 22.20 ilustra o *buffer* de recepção. Observe que o próximo *byte* a ser utilizado pelo processo receptor é o *byte* na posição 194. O receptor espera receber o *byte* na posição 200 do transmissor (o qual foi enviado, mas ainda não foi recebido). Quantos *bytes* a mais o receptor pode armazenar? Se o tamanho total do *buffer* receptor é N e existirem M posições ocupadas, então $N - M$ representa a quantidade de *bytes* permitidos pelo *buffer* receptor. Este valor é denominado **Janela de recepção**. Por exemplo, se $N = 13$ e $M = 6$, o tamanho da janela receptora é 7.



Figura 22.20 Janela de recepção.

Janela de Transmissão

O controle de fluxo pode ser implementado se o transmissor estabelecer uma janela – a **janela de transmissão** – com um tamanho menor do que ou igual ao tamanho da janela de recepção. Esta janela inclui os *bytes* enviados, mas não confirmados, e aqueles aguardando a transmissão. A Figura 22.21 ilustra a janela no *buffer* de transmissão.

Observe que o tamanho da janela de transmissão é igual ao tamanho da janela de recepção (no caso, tamanho igual a 7). Entretanto, isso não significa que o transmissor pode enviar mais 7 *bytes*; de fato, ele só poderá enviar mais 4 *bytes* porque 3 *bytes* já foram transmitidos. Perceba também que embora os *bytes* 207 a 211 estejam no *buffer*, eles não podem ser transmitidos até que os *bytes* a caminho cheguem ao receptor.

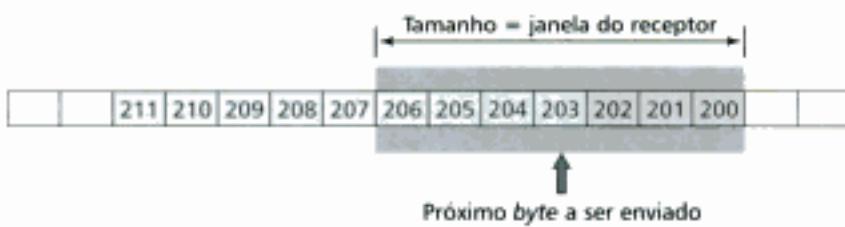


Figura 22.21 Buffer e janela de transmissão.

Janela de Transmissão Móvel

Vamos ver como as mensagens do receptor modificam a posição da janela de transmissão. No exemplo anterior, suponha que o transmissor envie mais 2 *bytes* e uma confirmação seja recebida do receptor (esperando o *byte* 203) sem nenhuma mudança no tamanho da janela de recepção (ou seja, a janela continua a ter 7 *bytes*). Nesse caso, o transmissor pode mover a janela e as posições ocupadas pelos *bytes* 200 a 202 podem ser recicladas. A Figura 22.22 ilustra a posição do *buffer* e a janela de transmissão antes e após este evento. Na parte b da figura, o transmissor pode enviar os *bytes* 205 a 209 (mais 5 *bytes*).

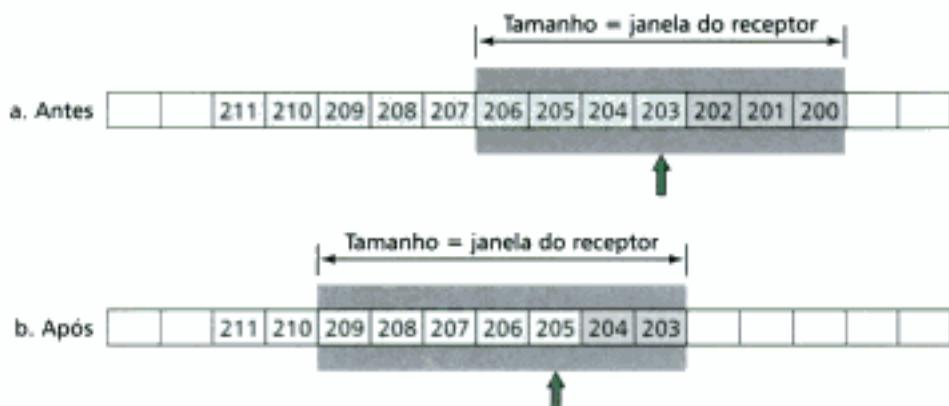


Figura 22.22 Janela de transmissão deslizante.

Expandindo a Janela de Transmissão

Se o processo que estiver recebendo dados não puder utilizá-los a uma velocidade superior à taxa de recepção, o tamanho da janela de recepção pode ser expandida (o *buffer* adquire novas posições vazias). Esta situação pode ser informada ao transmissor, resultando no aumento (expansão) do tamanho da janela. Na Figura 22.23, o receptor confirmou a recepção de mais 2 *bytes* (passando a esperar o *byte* na posição 205) e ao mesmo tempo aumentou o tamanho da janela de recepção para 10. Neste meio tempo, o processo transmissor criou mais 4 *bytes* e o protocolo TCP do transmissor enviou 5 *bytes*.

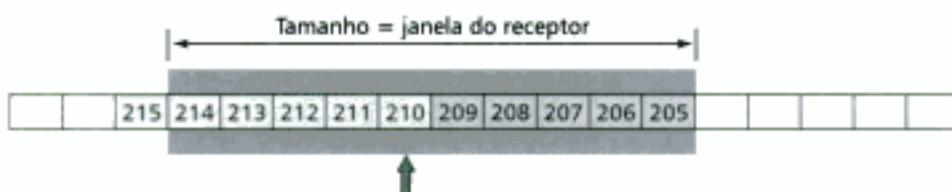


Figura 22.23 Expandindo a janela de transmissão.

Reduzindo a Janela de Transmissão

Se o processo que estiver recebendo consumir os dados mais lentamente do que eles estão sendo recebidos no *buffer*, o tamanho da janela de recepção é reduzido. Nesse caso, o receptor deve informar ao transmissor para que ele reduza o tamanho da janela de transmissão. Na Figura 22.24, o receptor recebeu os 5 *bytes* (205 a 209). Contudo, o processo receptor utilizou somente 1 *byte*, o que representa uma redução nas posições livres do *buffer* para 6 (10 - 5 + 1). Ele confirma os *bytes* nas posições 205 a 209 (esperando 210), mas também informa ao transmissor que reduza o tamanho da janela e não envie mais que 6 *bytes* novos. De fato, se o transmissor tiver enviado mais 2 *bytes* ao receptor, quando ele receber a notícia, e se tiver recebido mais 3 *bytes* do processo transmissor, fica configurada a situação mostrada na Figura 22.24.

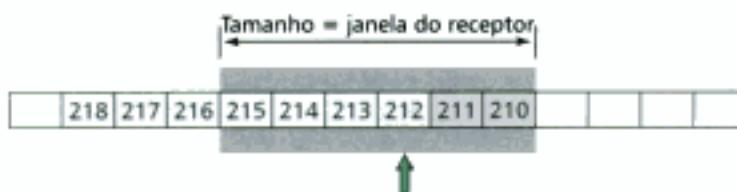


Figura 22.24 Reduzindo a janela de transmissão.

Fechando a Janela de Transmissão

O que acontece se o *buffer* receptor atingir a capacidade máxima? Nesse caso, o valor da janela de recepção vai a zero. Quando isto é informado ao transmissor, ele fecha a janela. O transmissor não pode enviar outro(s) *byte*(s) adicional(is) até que o receptor retorne um valor diferente de zero para a janela de recepção. Discutiremos esta questão novamente quando falarmos sobre a temporização (os *timers*) TCP.

No TCP, o tamanho da janela de transmissão é totalmente controlado pelo valor da janela de recepção (a quantidade de entradas livres no *buffer* do receptor). Contudo, o tamanho real da janela pode ser menor se houver congestionamento na rede.

Algumas observações sobre o esquema de janela móvel TCP:

- A origem não precisa enviar um segmento no valor total da janela de dados.
 - O tamanho da janela pode ser aumentado ou diminuído pelo *host* de destino.
 - O *host* de destino pode transmitir um ACK em um instante qualquer de tempo.
-

Síndrome da Janela Boba

Um problema sério que acontece na operação da janela móvel quando a aplicação transmissora produz dados lentamente ou a aplicação receptora consome os dados lentamente ou ambos. Qualquer uma dessas situações resulta na transmissão de dados em segmentos de tamanho muito pequenos, o que reduz a eficiência da operação. Por exemplo, se o TCP envia segmentos contendo somente 1 *byte* de dados, o datagrama terá 41 *bytes* (20 *bytes* para o cabeçalho TCP, 20 *bytes* para o cabeçalho IP e 1 *byte* de dado) e transfere somente 1 *byte* de dados do usuário. Nesse caso, o *overhead* vale 41/1, indicando que a capacidade da rede está sendo utilizada muito inefficientemente. Este problema é denominado **síndrome da janela boba**. Descreveremos como o problema é criado em cada uma das extremidades e então apresentaremos uma solução.

Síndrome Criada pelo Transmissor

O TCP transmissor pode gerar uma janela boba se estiver servindo uma aplicação que produz dados muito lentamente, por exemplo, 1 *byte* por vez. A aplicação escreve 1 *byte* por vez no *buffer* do protocolo TCP transmissor. Se o TCP transmissor não tiver nenhuma instrução específica ele pode criar um segmento de apenas 1 *byte* de dados. O resultado será um segmento de apenas 41 *bytes* viajando através de uma internet.

A solução é evitar que o TCP transmissor transmita dados no esquema de 1 *byte* por vez. O TCP transmissor deve ser forçado a coletar dados em quantidade antes de iniciar uma transmissão. Quanto tempo o TCP transmissor deve esperar? Se ele esperar demais, pode ocorrer atraso no processo. Se o TCP não esperar tempo suficiente, a ineficiência provocada pela transmissão de pequenos segmentos perdurará. Nagle apresentou uma solução elegante para esse problema.

Algoritmo de Nagle O algoritmo de Nagle é muito simples, mas resolve o problema elegantemente. Este algoritmo é aplicável ao TCP transmissor:

1. O TCP transmissor envia a primeira parte dos dados que ele recebe da aplicação transmissora (até mesmo se ela tiver apenas 1 *byte*).
2. Transmitido o primeiro segmento, o TCP transmissor acumula dados no *buffer* de saída e espera o TCP receptor enviar uma confirmação ou serem acumulados dados em quantidade suficiente para esgotar a capacidade máxima de um segmento. Nesse momento, o TCP transmissor está autorizado a enviar o próximo segmento.
3. A etapa 2 é repetida para o resto da transmissão. O segmento 3 deve ser enviado se uma confirmação for recebida para o segmento 2 ou se dados suficientes são acumulados e estão esgotando o tamanho máximo do segmento.

A elegância do algoritmo de Nagle reside na simplicidade, como ele trata o problema, e no gerenciamento da velocidade de produção de dados pelo processo transmissor, em relação à velocidade da rede de dados. Se a taxa de produção de dados pela aplicação transmissora for maior que a velocidade da rede, os segmentos serão maiores (segmentos de tamanho máximo). Se a taxa de produção de dados pela aplicação for menor que a velocidade da rede, os segmentos são menores (menores que o tamanho máximo do segmento).

Síndrome Criada Pelo Receptor

O TCP receptor pode gerar uma janela boba se estiver servindo uma aplicação que produz dados muito lentamente, por exemplo, 1 byte por vez. Suponha que aplicação transmissora crie dados em blocos de 1k, mas a aplicação receptora consegue utilizar apenas 1 byte por vez. Suponha também que o buffer de entrada do TCP receptor possua um tamanho de 4k. O transmissor envia os primeiros 4 kbytes de dados. O receptor armazena-os no buffer. Nesse caso, o buffer é utilizado completamente. O TCP receptor informa ao transmissor um tamanho de janela igual a zero, significando que o transmissor deve cessar a transmissão de dados imediatamente. A aplicação receptora lê o primeiro byte de dados do buffer de entrada do TCP receptor. Nesse caso, 1 byte de espaço é aberto no buffer de chegada. O TCP receptor informa ao transmissor um tamanho de janela igual a 1 byte. O TCP transmissor, esperando impacientemente para enviar dados, toma isso como uma notícia boa e envia um segmento transportando somente 1 byte. O procedimento continuará. Um byte de dados é utilizado pela aplicação receptora e um segmento transportando 1 byte de dados é enviado pelo transmissor. Novamente, temos um problema de ineficiência e a criação de uma síndrome.

Duas soluções foram propostas para evitar a síndrome da janela boba criada pela aplicação que faz uso lento desses dados.

Solução de Clark A solução de Clark foi enviar uma confirmação tão logo os dados sejam recebidos, mas informando um tamanho de janela igual a zero até que haja espaço suficiente para acomodar um segmento de tamanho máximo ou metade do buffer esteja vazio.

Atraso da Confirmação A segunda solução é atrasar o envio da confirmação. Isto significa que, ao receber um segmento, não será produzida uma resposta de confirmação imediatamente. O receptor espera até que haja uma quantidade de espaço decente no buffer de entrada do receptor antes de confirmar o recebimento dos segmentos. O atraso da confirmação evita que o TCP transmissor move a janela de transmissão. Após ter enviado os dados na janela, ele pára. Isto resolve a síndrome.

O atraso da confirmação também possui outra vantagem: o atraso reduz o tráfego. O receptor não responde com uma confirmação para cada segmento. Contudo, há também uma desvantagem: o atraso pode forçar o transmissor a retransmitir os segmentos sem confirmação.

O protocolo consegue balancear vantagens e desvantagens especificando um atraso de confirmação máximo de 500 ms.

Controle de Erros

Mencionamos diversas vezes que o TCP é um protocolo de transporte confiável. Isto significa que um processo da camada de aplicação confia no TCP para entregar todos os dados em ordem à aplicação na outra extremidade, sem erros, isto é, sem partes perdidas ou duplicadas. O controle de erros do TCP inclui a detecção de segmentos corrompidos, perdidos, fora de ordem e segmentos duplicados.

O TCP usa três ferramentas simples: *checksum*, ACK e *time-out*. Todo segmento inclui um campo *checksum* utilizado para verificar se o segmento foi corrompido. Se o segmento foi de fato corrompido deve ser descartado pelo protocolo TCP de destino. O TCP usa as respostas ACK para confirmar a recepção daqueles segmentos que alcançaram o destino sem sofrer algum tipo de dano. Além disso, o TCP não usa respostas ACK negativas. Se um segmento não receber confirmação antes do *time-out*, é considerado corrompido ou perdido.

O TCP do host de origem dispara um contador *time-out* para cada segmento transmitido. Todos os contadores são verificados periodicamente. Quando um contador expira, o segmento correspondente é considerado perdido ou danificado, sendo solicitada a retransmissão dele.

Hidden page

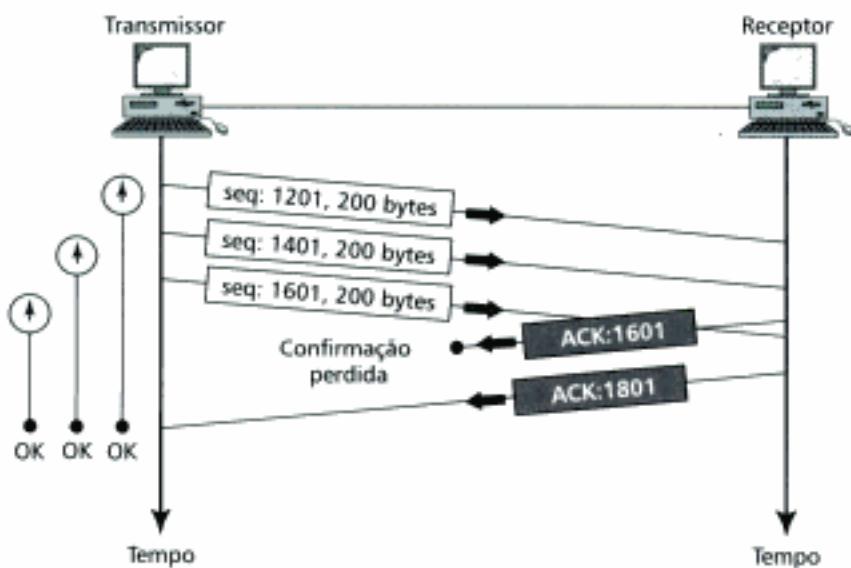


Figura 22.26 Confirmação (ACK) perdida.

byte especificado, através do número de confirmação. Por exemplo, se o host de destino envia um segmento ACK com um número de confirmação para o byte na posição 1801, fica confirmado a recepção dos bytes de 1201 a 1800. Se o host de destino enviou previamente um segmento ACK para o byte na posição 1601, significa que ele recebeu os bytes de 1201 a 1600 e as perdas de ACKs são irrelevantes.

Relógios do TCP

O TCP utiliza quatro **relógios TCP (TCP timers)** para suavizar a operação, conforme ilustra a Figura 22.27.



Figura 22.27 Temporizadores (timers) TCP.

Relógio de Retransmissão

O TCP usa o **relógio de retransmissão** para controlar o tempo de retransmissão dos segmentos perdidos ou descartados, esperando o tempo de confirmação de um segmento. Quando o TCP transmite um segmento, ele configura um relógio de retransmissão para esse segmento particular. Pode ocorrer duas situações:

1. Se um ACK é recebido para um segmento particular antes do relógio expirar, o relógio é ignorado.
2. Se ocorrer um *time-out* antes do ACK ser recebido, o segmento é retransmitido e o relógio é resetado.

Cálculo do Tempo de Retransmissão Sabemos que o TCP é protocolo da camada de transporte. Toda conexão TCP envolve duas partes, que podem estar na mesma rede física ou então estar localizadas muito afastados um do outro (por exemplo, em lados opostos do globo terrestre). Além

Hidden page

Para corrigir este conflito, o TCP usa um **relógio de persistência** para cada conexão. Quando o TCP transmissor recebe um ACK com um tamanho de janela igual a zero ele dispara um relógio de persistência. Quando o contador de persistência expira, o TCP transmissor envia um segmento especial de sondagem denominado *probe*. Este segmento possui somente 1 byte de dados. Ele tem um número de seqüência, mas o número de seqüência nunca é confirmado. Ele é ignorado até mesmo no cálculo do número de seqüência para o restante dos dados. O segmento *probe* alerta o TCP receptor que a confirmação foi perdida ou deveria ser retransmitida.

O valor do relógio de persistência é identificado com o valor do tempo de retransmissão. Entretanto, se uma resposta não for recebida do receptor, outro segmento de sondagem é enviado e o valor do relógio de persistência é duplicado e resetado. O transmissor continua a enviar os segmentos de sondagem, a duplicar e a resetar o relógio de persistência até que o valor alcance um limiar (tipicamente 60s). Após esse limite, o transmissor envia um segmento de sondagem a cada 60 s até que a janela seja reaberta.

Relógio Keep-Alive

Um **relógio keep-alive** é utilizado em algumas implementações para evitar conexões ociosas entre dois protocolos TCPs. Suponha que um cliente abra uma conexão TCP com um servidor, transfere uma certa quantidade de dados e fique em silêncio a partir de então. Talvez, o cliente tenha travado. Nesse caso, a conexão permanece aberta indefinidamente.

Para remediar esta situação, muitas implementações oferecem a um servidor um relógio *keep-alive*. Esse relógio funciona assim: o relógio é resetado toda vez que um servidor escuta um cliente. O valor do *time-out* geralmente é 2h. Um segmento de sondagem é transmitido pelo servidor, caso ele não escute o cliente após 2h. Se não houver uma resposta após 10 sondagens, cada qual é separada de 75s da outra, é assumido que o cliente está inativo e a conexão é terminada.

Relógio de Espera

O **relógio de espera** é utilizado durante o término da conexão. Quando o TCP fecha uma conexão, ele não considera a conexão fechada efetivamente. A conexão é mantida no limbo por um período de espera. Isto possibilita que os segmentos FIN duplicados, se existirem, cheguem ao destino para serem descartados. O valor para este relógio é usualmente 2 vezes o tempo de vida esperado de um segmento.

Controle de Congestionamento

Discutiremos o controle de congestionamento do protocolo TCP no próximo capítulo.

Outras Características

Há outras duas características TCP que precisamos discutir: *pushing* de dados e o controle da urgência dos dados.

Pushing de dados

Vimos que o TCP transmissor usa um *buffer* para armazenar o fluxo de dados oriundo da aplicação transmissora. O TCP transmissor pode determinar o tamanho dos segmentos. No lado receptor, o TCP também utiliza um *buffer* para acomodar os dados, quando recebidos, de modo a entregá-los ao processo receptor quando a aplicação estiver pronta para receber ou quando o TCP receptor julgar que é conveniente entregar. Este tipo de flexibilidade aumenta a eficiência do protocolo TCP.

Entretanto, há ocasiões nas quais a aplicação não se sente "confortável" com este tipo de flexibilidade. Por exemplo, considere uma aplicação em um *host* comunicando interativamente com outra aplicação (noutro *host*). A aplicação transmissora deseja enviar um comando à aplicação do

outro lado e receber uma resposta imediatamente. O atraso do processo de transmissão de dados pode não ser aceitável à aplicação que necessitar da resposta.

O TCP está preparado para controlar este tipo de situação. A aplicação que estiver enviando de um lado solicita uma operação de *push de dados*. Isto significa que o TCP transmissor não deve esperar a janela ser preenchida. O TCP deve criar um segmento e enviá-lo imediatamente. O TCP transmissor também pode configurar o *bit push* (PSH) para informar ao TCP receptor que o segmento inclui dados que devem ser entregues à aplicação receptora tão logo quanto possível e não esperar mais dados chegarem ao *buffer*.

Embora a operação de *pushing* possa ser requisitada por uma aplicação, hoje em dia a maioria das implementações ignoram tais solicitações. O TCP pode escolher se deseja usar esta operação.

Dados com Urgência

Sabemos que o TCP é um protocolo orientado a fluxo. Recordemos que isso significa que os dados são passados da aplicação ao TCP como uma cadeia de caracteres. Cada *byte* de dados ocupa uma posição na cadeia. Contudo, existem ocasiões nas quais uma determinada aplicação necessita enviar os *bytes urgentes*. A aplicação transmissora deseja que uma porção dos dados seja lido fora da ordem da fila pela aplicação receptora. Suponha que a aplicação transmissora esteja enviando uma massa de dados para serem processados pela aplicação receptora. Quando o resultado do processamento retornar, a aplicação transmissora verifica que está tudo errado. Ela deseja abortar o processo, mas a quantidade de dados enviada foi imensa. Se ocorrer um comando abortar (control + C), estes caracteres serão armazenados e transmitidos às últimas posições do *buffer* do protocolo TCP receptor. Eles serão entregues à aplicação receptora após todo o processamento de dados ter ocorrido.

Nesse caso, a solução é enviar um segmento com o *bit URG* configurado. A aplicação transmissora informa ao TCP transmissor que aquela porção contém dados urgentes. O protocolo TCP transmissor cria um segmento e insere o indicador de urgência de dados no início do segmento. O resto do segmento pode conter dados normais do *buffer*. O campo indicador de urgência no cabeçalho define onde termina os dados urgentes e onde começa os dados normais.

Quando o TCP receptor recebe um segmento com o *bit URG* configurado, ele extrai os dados urgentes do segmento, usando o valor do indicador de urgência, e o entrega (fora de ordem) à aplicação receptora.

22.4 TERMOS-CHAVE

Cliente	Paradigma cliente-servidor
Comunicação entre processos	Relógio de persistência
Controle de fluxo	Relógio de retransmissão
Datagrama UDP	Relógio do período de espera
Diagrama de transição de estados	Relógio <i>keep-alive</i>
Endereço de <i>socket</i>	Relógio TCP
Estabelecimento da conexão	Round-Trip Time (RTT)
<i>Handshake triplo (three-way handshake)</i>	Segmento
Janela móvel (janela deslizante)	Serviço orientado à conexão
Janela receptora	Serviço sem conexão
Janela transmissora	Servidor
Máquina de estado finito	Síndrome da janela boba
Número de porta	Término da conexão
Número de porta permanente	Transmission Control Protocol (TCP)
Número de porta temporário	User Datagram Protocol (UDP)
Número de seqüência	

22.5 RESUMO

- UDP e TCP são protocolos da camada de transporte que controlam a comunicação entre processos finais.
- O UDP é um protocolo não confiável e sem conexão que gera pouco *overhead* e oferece entrega rápida de dados entre processos.
- No paradigma cliente-servidor, uma aplicação rodando em um *host* local, denominado cliente, necessita dos serviços de uma aplicação que roda em um *host* remoto, denominado servidor.
- Toda aplicação possui um único número de porta para distingui-la de outras aplicações que estiverem rodando ao mesmo tempo na máquina.
- Para a aplicação cliente é atribuído um número de porta aleatório denominado número de porta temporário.
- Para a aplicação servidora é atribuído um número de porta universal denominado número de porta permanente.
- A combinação número de porta e endereço IP é denominada endereço *socket* que define univocamente um processo e um *host*.
- O pacote UDP é denominado datagrama UDP.
- O UDP não oferece controle de fluxo.
- O Transmission Control Protocol (TCP) é um protocolo de transporte orientado à conexão e confiável utilizado no modelo da Internet.
- A unidade de dados transferida entre dois dispositivos usando TCP é denominada segmento. Um segmento TCP tem de 20 a 60 bytes de cabeçalho seguido dos dados da aplicação (o *payload*).
- O TCP usa um mecanismo de janela móvel para controlar o fluxo de dados.
- A detecção de erros é controlada no TCP através do *checksum*, ACK e *time-out*.
- No TCP, os segmentos corrompidos ou perdidos são retransmitidos e os segmentos duplicados são descartados.
- O TCP possui quatro relógios (*timers*). São eles: retransmissão, persistência, *keep-alive* e espera.
- A fase de estabelecimento da conexão requer três etapas; a fase do término da conexão (desconexão) normalmente acontece em quatro etapas.
- O TCP é implementado como uma máquina de estados finitos.
- O tamanho da janela no TCP é determinado pelo receptor.

22.6 PRATIQUE OS CONHECIMENTOS ADQUIRIDOS

Questões de Revisão

1. Qual é a diferença entre a comunicação entre processos finais e a comunicação entre *hosts* (*host-to-host*)?
2. Como uma porta permanente difere de uma porta temporária?
3. O que é endereço de *socket*?
4. Quando o *handshake* triplo (*three-way handshake*) é utilizado?
5. Por que uma aplicação usaria o UDP se o TCP é mais completo?
6. Como é denominado um pacote UDP? E o pacote TCP?
7. Qual é o propósito do número de sequência em um pacote TCP?
8. Qual é o propósito do controle de fluxo?
9. O que é a síndrome da janela boba?
10. Qual é o propósito do algoritmo de Nagle?
11. Que métodos conseguem evitar a síndrome da janela boba criada pelo receptor?
12. Cite os relógios (*timers*) TCP.
13. Qual é a finalidade da operação de *pushing* de dados no TCP?
14. Como o TCP pode controlar os dados urgentes?

Questões de Múltipla Escolha

15. UDP e TCP são ambos protocolo da camada _____.
 - Física
 - De enlace
 - De rede
 - De transporte
16. Qual das seguintes funções é realizada pelo UDP?
 - Comunicação entre processos finais
 - Comunicação *host-to-host*
 - Confiabilidade (fim a fim) de entrega de dados
 - Todas opções representam funções do UDP
17. O UDP necessita do endereço de _____ para realizar as entregas de datagramas UDP para a aplicação correta.
 - Porta
 - Aplicação
 - Internet
 - Enlace (MAC)
18. Qual das seguintes opções representa um número de porta válido?
 - 0
 - 513
 - 65.535
 - Todas as opções apresentam números válidos
19. A definição de confiabilidade inclui _____.
 - Entrega livre de erros
 - Recebimento da mensagem completa
 - Entrega na ordem
 - Todas opções anteriores
20. Qual da seguintes opções é encontrada no UDP?
 - Número de seqüência para cada datagrama UDP
 - ACKs do transmissor
 - Controle de fluxo
 - Nenhuma das opções anteriores
21. O endereço da porta de origem do cabeçalho do datagrama UDP define _____.
 - O *host* transmissor
 - O *host* receptor
22. Qual das seguintes opções *não* faz parte do cabeçalho do datagrama UDP?
 - Tamanho do cabeçalho
 - Endereço da porta de origem
 - Checksum*
 - Endereço da porta de destino
23. O _____ define a aplicação cliente.
 - Número de porta temporário
 - Endereço IP
 - Número de porta permanente
 - Endereço físico
24. O _____ define a aplicação servidora.
 - Número de porta temporário
 - Endereço IP
 - Número de porta permanente
 - Endereço físico
25. O IP é responsável pela comunicação _____ enquanto que o TCP é responsável pela comunicação _____.
 - Entre *hosts*; entre processos
 - Entre processos; entre *hosts*
 - Entre processos; entre nós
 - Entre nós; entre processos
26. Um *host* pode ser identificado através de _____ enquanto a aplicação que estiver rodando no *host* pode ser identificada por _____.
 - Um endereço IP; um número de porta
 - Um número de porta; um endereço IP
 - Um endereço IP; um endereço de *host*
 - Um endereço IP; um número de porta permanente
27. O endereço _____ identifica univocamente a aplicação que estiver rodando.
 - IP
 - De *host*
 - NIC
 - Socket

28. O campo _____ é usado para ordenar os pacotes de uma mensagem.
- Indicador de urgência
 - Checksum*
 - Número de seqüência
 - Número ACK
29. O campo _____ é usado para detecção de erros.
- Indicador de urgência
 - Checksum*
 - Número de seqüência
 - Número ACK
30. Multiplicando o tamanho do campo cabeçalho por _____ determinamos a quantidade total de *bytes* presentes no cabeçalho TCP.
- 2
 - 4
 - 6
 - 8
31. Dados urgentes requerem o campo indicador de urgência assim como o bit URG no campo _____.
- Controle
 - Offset*
 - Número de seqüência
 - Reservado
32. Na _____, dados são enviados ou processados muito inefficientemente, tal como 1 *byte* por vez.
- Síndrome de Nagle
 - Síndrome da janela boba
 - Síndrome da janela móvel
 - Confirmação atrasada
33. Para evitar a síndrome da janela criada por um receptor que processa dados muito lentamente, podemos utilizar o(a) _____.
- Solução de Clark
 - Algoritmo de Nagle
 - Confirmação atrasada
 - (a) ou (c)
34. Para evitar a síndrome da janela criada por um transmissor que envia dados a uma taxa muito baixa, podemos utilizar o(a) _____.
- Solução de Clark
 - Algoritmo de Nagle
 - Confirmação atrasada
 - (a) ou (c)
35. Uma resposta ACK igual 1000 sempre significa que _____.
- 999 *bytes* foram recebidos com sucesso
 - 1000 *bytes* foram recebidos com sucesso
 - 1001 *bytes* foram recebidos com sucesso
 - Nenhuma das respostas anteriores
36. O relógio _____ evita um longo tempo de ociosidade entre dois protocolos TCPs.
- De retransmissão
 - De persistência
 - Keep-alive*
 - De espera
37. O relógio _____ é necessário para controlar a informação sobre o tamanho zero da janela.
- De retransmissão
 - De persistência
 - Keep-alive*
 - De espera
38. O algoritmo de Karn é utilizado nos cálculos pelo relógio _____.
- De retransmissão
 - De persistência
 - Keep-alive*
 - De espera
39. O relógio _____ é usado na fase de término da conexão.
- De retransmissão
 - De persistência
 - Keep-alive*
 - De espera
40. O relógio _____ mantém-se informado sobre o tempo entre a transmissão de um segmento e a recepção de uma resposta de confirmação.
- De retransmissão
 - De persistência
 - Keep-alive*
 - De espera
41. O estabelecimento da conexão envolve um *handshake* _____.
- Único
 - Duplo
 - Triplô
 - Nenhuma das respostas anteriores
42. Um segmento especial de sondagem, denominado probe, é enviado pelo protocolo TCP transmissor quando o relógio _____ é finalizado.
- De transmissão
 - De persistência
 - Keep-alive*
 - De espera

Exercícios

43. Nos casos onde a confiabilidade dos processos não tem muita importância, o UDP seria uma boa escolha como protocolo da camada de transporte. Dê exemplos de casos específicos.
44. O IP e o UDP são não confiáveis no mesmo grau? Por quê?
45. Os números de porta precisam ser únicos? Por quê? Por que os números de porta são menores que os endereços IP?
46. Qual é a definição que o dicionário dá para a palavra *efêmero*? Pode a definição do dicionário ser aplicada ao número de porta temporário?
47. Qual é o tamanho mínimo de um datagrama UDP? E o tamanho máximo?
48. Qual é o tamanho mínimo dos dados do processo final que pode ser encapsulado num datagrama UDP? E qual é o tamanho máximo?
49. Um cliente usa o protocolo UDP para transmitir dados a um servidor. Os dados têm 16 bytes. Determine a eficiência desta transmissão, isto é, a razão entre bytes úteis (*payload*) e total de bytes.
50. Repita o exercício 49 calculando a eficiência da transmissão no nível da camada de rede, isto é, do protocolo IP. Assuma que o cabeçalho IP não possui opções.
51. Repita o exercício 48 calculando a eficiência da transmissão no nível da camada de enlace. Assuma que o cabeçalho IP não possui opções e use o padrão Ethernet na camada de enlace.
52. Qual é o tamanho máximo do cabeçalho TCP? E o tamanho mínimo?
53. Se o valor do campo HLEN é 0111, quantos bytes de opção são incluídos no segmento?
54. O que você pode dizer sobre o segmento TCP onde o campo controle possui um dos seguintes valores?
 - a. 000000
 - b. 000001
 - c. 010001
 - d. 000100
 - e. 000010
 - f. 010010
55. O TCP está transmitindo dados a 1 megabyte por segundo (8 Mbits/s). Se o número de seqüência começa no 7000, quanto tempo leva antes do número de seqüência voltar a zero?
56. Uma conexão TCP está usando um tamanho de janela igual a 10.000 bytes e o número da resposta de confirmação anterior foi 22.001. O TCP recebe um segmento com número ACK igual 24.001. Desenhe um diagrama para mostrar a situação da janela antes e depois.
57. Repita o exercício 56 se o receptor modificou o tamanho da janela para 11.000.
58. Repita o exercício 56 se o receptor modificou o tamanho da janela para 90.000.
59. Um cliente usa TCP para enviar dados a um servidor. Os dados têm 16 bytes. Determine a eficiência desta transmissão, isto é, a razão entre bytes úteis (*payload*) e total de bytes.
60. Repita o exercício 49 calculando a eficiência da transmissão no nível da camada de rede, isto é, do protocolo IP. Assuma que o cabeçalho IP não possui opções.
61. Repita o exercício 48 calculando a eficiência da transmissão no nível da camada de enlace. Assuma que o cabeçalho IP não possui opções e use o padrão Ethernet na camada de enlace.

Controle de Congestionamento e Qualidade de Serviço

Controle de congestionamento e qualidade de serviço são duas questões tão próximas que melhorar uma significa melhorar a outra e ignorar uma geralmente significa ignorar a outra. Muitas das técnicas para evitar ou eliminar o congestionamento também melhoraram a qualidade de serviço.

Postergamos essa discussão até o presente capítulo porque essas questões estão relacionadas não somente a uma camada, mas envolvem três camadas: enlace, rede e transporte. Esperamos até aqui para podermos discutir estas questões uma única vez, ao invés de repetir o assunto três vezes. Ao longo deste capítulo daremos exemplos de controle de congestionamento e qualidade de serviços nas diferentes camadas.

23.1 TRÁFEGO DE DADOS

O foco principal do controle de congestionamento e da qualidade de serviço é o tráfego de dados. No controle de congestionamento tentamos evitar o congestionamento do tráfego. Na qualidade de serviço tentamos criar um ambiente apropriado ao tráfego. Assim, discutiremos as questões relativas ao tráfego de dados antes de iniciar o estudo do controle de congestionamento e da qualidade de serviço.

Descritores do Tráfego

A terminologia do tráfego apresenta definições e valores qualitativos que representam o fluxo de dados. A Figura 23.1 ilustra um fluxo de dados e algumas das caracterizações típicas.

Taxa Média de Dados

A **taxa média de dados** é definida como a quantidade de *bits* enviados dividida pelo intervalo de tempo decorrido durante a transmissão. A fórmula abaixo mostra essa mesma definição matematicamente:

$$\text{Taxa média de dados} = \frac{\text{quantidade de dados}}{\text{tempo}}$$

A taxa média de transmissão de dados é uma característica muito útil porque sugere a largura de banda média necessária ao tráfego.



Figura 23.1 Descritores do tráfego.

Taxa Máxima de Dados

A **taxa máxima de dados** define o valor de pico da taxa de dados. Na Figura 23.1, a taxa máxima é o maior valor no eixo y. A taxa máxima é uma medida muito importante porque indica a largura de banda máxima que uma rede deve suportar de modo a manter o tráfego através da rede sem variações perceptíveis no fluxo de dados.

Tamanho Máximo da Rajada

Embora a taxa máxima represente um valor crítico para a rede, geralmente podemos ignorá-la se o máximo acontecer instantaneamente. Por exemplo, se dados estão fluindo a uma taxa fixa de 1 Mbps e repentinamente a taxa salta para 2 Mbps, durante 1ms, a rede provavelmente pode controlar a situação. Entretanto, se o pico durar uns 60 ms pode ocorrer um problema na rede. O **tamanho máximo da rajada** normalmente refere-se ao intervalo de tempo que o tráfego consegue permanecer na taxa máxima.

Largura de Banda Efetiva

A **largura de banda efetiva** é a largura de banda que a rede deve possuir para que o tráfego flua. A largura de banda efetiva é uma função dos parâmetros anteriores: taxa média, taxa máxima e tamanho máximo da rajada de dados. O cálculo desse parâmetro pode ser muito complexo.

Perfis do Tráfego

Para os propósitos deste livro, um fluxo de dados pode acontecer em um dos seguintes perfis: taxa constante de *bits*, taxa variável de *bits* ou em rajadas. Deve ser observado que os dois últimos perfis muitas vezes são tratados como um único perfil.

Taxa Constante de Bits

O modelo de tráfego de dados baseado na **taxa constante de bits (CBR)** estabelece que o fluxo acontece a uma taxa fixa (constante). Neste tipo de fluxo, a taxa média é idêntica à taxa máxima de dados. A rajada de dados não é aplicável. Este tipo de tráfego é muito fácil de ser controlado por uma rede desde que o tráfego seja previsível. Assim, a rede tem como saber antecipadamente que valor de largura de banda alocar para este tipo de fluxo. A Figura 23.2 ilustra o tráfego à taxa constante de *bits*.



Figura 23.2 Tráfego à taxa constante de *bits*.

Taxa Variável de Bits

No tráfego a uma **taxa variável de bits** (VBR), a taxa de dados varia suavemente no tempo ao invés de sofrer mudanças abruptas. Neste tipo de fluxo, a taxa média e a taxa máxima de dados são diferentes. O tamanho da rajada de dados é usualmente muito pequeno. Este tipo de tráfego é muito mais difícil de ser controlado que o tráfego à taxa constante, mas ele normalmente não precisa, como veremos, ser modelado. A Figura 23.3 mostra um tráfego a uma taxa variável de bits.

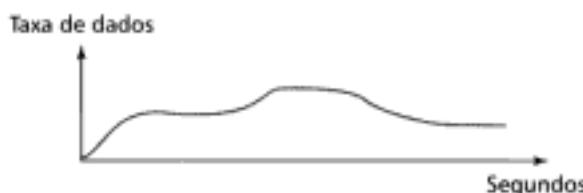


Figura 23.3 Tráfego à taxa variável de bits.

Rajadas

No tráfego em **rajada de dados**, dados são transmitidos abrupta e violentamente durante um curto intervalo de tempo. Pode ocorrer inclusive casos extremos onde a taxa de transmissão salta de zero para um determinado valor, por exemplo para 1 Mbps, em poucos microsegundos. A recíproca também é verdadeira, a taxa de dados pode mudar repentinamente do valor da rajada para zero. As taxas média e máxima de transmissão são muito diferentes nesse tipo de fluxo. O tamanho máximo da rajada é bastante significativo. Este tipo de tráfego é o mais difícil de controlar numa rede porque o perfil é bastante imprevisível. Para controlar este tipo de tráfego, a rede normalmente precisa reformatá-lo, usando alguma técnica de modelamento do tráfego, como discutiremos mais adiante. O tráfego em rajadas é uma das principais causas do congestionamento numa rede. A Figura 23.4 ilustra o tráfego em rajadas.

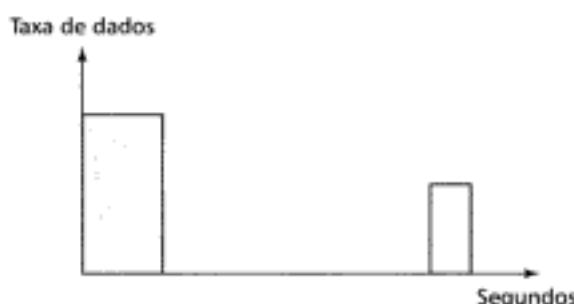


Figura 23.4 Tráfego em rajadas.

23.2 CONGESTIONAMENTO

Uma questão importante numa rede de comutação de pacotes é o **congestionamento**. Congestionamento numa rede pode ocorrer se a **carga** na rede, a quantidade de pacotes enviados para a rede, superar a **capacidade** da rede, isto é, a quantidade de pacotes que uma rede consegue controlar. O **controle de congestionamento** trata das técnicas e dos mecanismos de controle para manter a carga abaixo da capacidade da rede.

A pergunta essencial talvez seja: por que existe congestionamento na rede? O congestionamento ocorre em qualquer sistema que envolve algum tipo de espera. Por exemplo, um congestionamento de trânsito acontece muitas vezes devido ao aumento anormal no fluxo de carros, provocado por algum evento inesperado, como um acidente, ou durante a hora do *rush*.

O congestionamento numa rede ou *internetworking* ocorre porque os roteadores e *switches* possuem filas de dados, *buffers* armazenando os pacotes antes e após o processamento. Um roteador, por exemplo, possui uma fila de pacotes na entrada e outra na saída de cada interface. Quando um pacote é recebido na interface de chegada, ele passa por três etapas antes de partir, conforme ilustra a Figura 23.5.

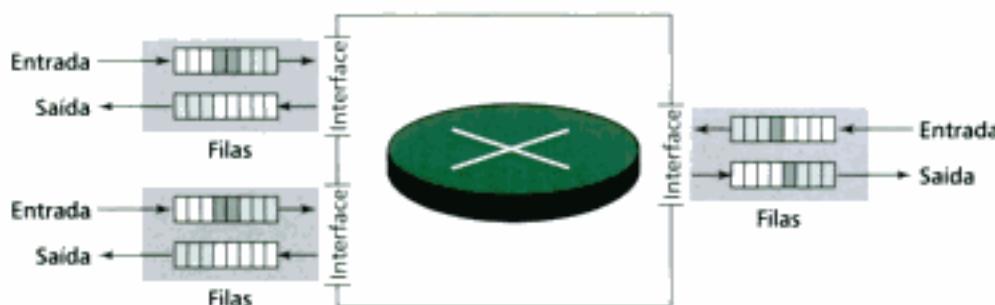


Figura 23.5 Pacotes de chegada.

1. O pacote é colocado na de fila de entrada enquanto aguarda verificação.
2. O módulo de processamento do roteador retira o pacote da fila de entrada toda vez que o pacote chega na posição final da fila. Em seguida, o roteador usa a tabela de roteamento e o endereço de destino no cabeçalho do pacote para determinar a rota.
3. O pacote é introduzido na fila de saída da interface apropriada, enquanto aguarda a transmissão.

Precisamos ficar atentos a duas questões. Primeira, se a taxa de chegada de pacotes na fila de entrada for maior que a taxa de processamento, a tendência será tornar o conteúdo da fila ainda maior (podendo, inclusive, esgotá-la). Segundo, se a taxa de transmissão dos pacotes na fila de saída for menor que a taxa de processamento dos pacotes, a fila de saída poderá ter a capacidade esgotada.

Performance da Rede

O controle de congestionamento envolve dois fatores que medem a performance de uma rede: **atraso (delay)** e **throughput**.

Atraso versus Carga

A Figura 23.6 ilustra a relação entre o atraso dos pacotes e a carga da rede.

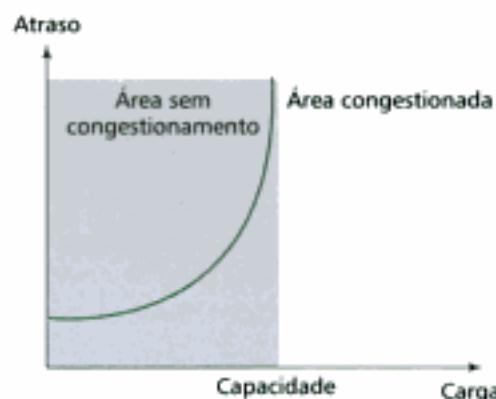


Figura 23.6 Atraso do pacote e carga da rede.

Observe que, quando a carga é muito menor que a capacidade da rede, o atraso é mínimo. Este atraso mínimo é provocado pelo atraso de propagação e atraso de processamento, sendo ambos desprezíveis. Contudo, quando a carga aproxima-se da capacidade da rede, o atraso total cresce assintoticamente visto que os tempos de espera nas filas (por exemplo, para todos roteadores no caminho) devem ser levados em consideração. Observe que o atraso tende a infinito quando a carga supera a capacidade da rede. Se isto não parecer razoável, considere o tamanho das filas formadas quando quase nenhum pacote consegue alcançar o destino ou estão alcançando o destino com atraso muito grande. O atraso produz um efeito negativo sobre a carga e, consequentemente, sobre o congestionamento. Quando um pacote atrasa, a origem não recebe a resposta de confirmação do destino e, muitas vezes, inicia a retransmissão do pacote, o que aumenta os atrasos na rede e o congestionamento piora.

Throughput versus Carga

Definimos o *throughput* no Capítulo 3 como a taxa de *bits* passando através de um ponto específico em certa unidade de tempo (geralmente um segundo). Podemos estender a definição substituindo *bits* por pacotes e ponto por rede. Assim, o *throughput* de uma rede é a quantidade de pacotes passando através de uma rede numa unidade de tempo. Podemos desenhar o *throughput versus a carga da rede*, conforme ilustra a Figura 23.7.

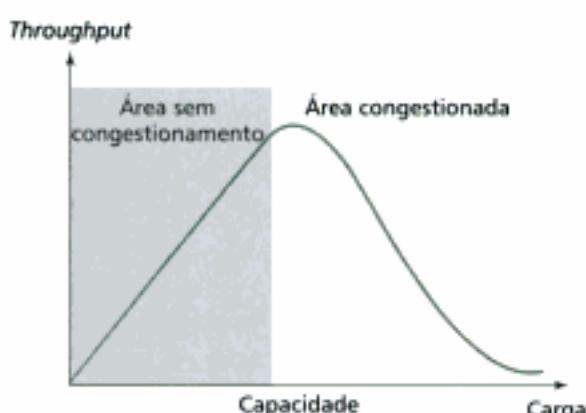


Figura 23.7 Throughput versus carga da rede.

Observe que, quando a carga está abaixo da capacidade da rede, o *throughput* aumenta proporcionalmente a cada nó. Esperamos que o *throughput* permaneça constante após a carga atingir a capacidade da rede, mas em vez disso o *throughput* começa a cair rapidamente. A explicação é a taxa de descarte de pacotes pelos roteadores. Quando a carga excede a capacidade da rede as filas são esgotadas e os roteadores têm que descartar alguns pacotes. O descarte de pacotes não reduz a quantidade de pacotes na rede porque os *hosts* de origem iniciam a retransmissão desses pacotes, usando os mecanismos de *time-out*, tão logo sintam a falta da resposta de confirmação dos *hosts* de destino.

23.3 CONTROLE DE CONGESTIONAMENTO

O controle de congestionamento trata as técnicas e mecanismos de prevenção (antes que aconteçam) e contenção (após terem acontecido) de congestionamento. Em geral, dividimos o controle de congestionamento em duas grandes categorias: controle em malha aberta (medida preventiva) e controle em malha fechada (medição de contenção).

Controle de Congestionamento em Malha Aberta

No **controle de congestionamento em malha aberta**, políticas são aplicadas para evitar os congestionamentos antes que aconteçam. A implementação desses mecanismos de controle é aplicada ao *host* de origem ou de destino. Abaixo listamos e descrevemos brevemente algumas políticas capazes de prevenir o congestionamento.

Política de Retransmissão

Uma boa política de retransmissão pode evitar o congestionamento. A política de retransmissão e os relógios de retransmissão devem ser ajustados de maneira a otimizar a eficiência e, ao mesmo tempo, evitar o congestionamento.

Política de Janela

O tipo de janela transmissora também pode afetar o congestionamento. A janela *Selective Repeat* é melhor que a janela Go-Back-N para controle de congestionamento.

Política de Confirmação

A política de confirmação imposta pelo receptor também pode afetar o congestionamento. Se o receptor não responder confirmado cada pacote recebido, ele pode reduzir a velocidade de transmissão do *host* de origem, prevenindo o congestionamento.

Política de Descarte

Uma boa política de descarte aplicada aos roteadores pode evitar o congestionamento, ao mesmo tempo, sem prejudicar a integridade da transmissão. Por exemplo, numa transmissão de áudio, se a política manda descartar os pacotes menos sensíveis quando o congestionamento está prestes a acontecer, a qualidade do som é ainda preservada e o congestionamento é prevenido.

Política de Admissão

Uma política de admissão, a qual é um mecanismo da qualidade de serviço, também pode evitar o congestionamento nas redes de circuitos virtuais. Os nós de comutação no fluxo verificam primeiramente os recursos requeridos pelo fluxo antes de admiti-lo na rede.

Controle de Congestionamento em Malha Fechada

Os mecanismos de **controle de congestionamento em malha fechada** tentam aliviar um congestionamento, após ter ocorrido de fato. Muitos mecanismos são usados por diferentes protocolos. Descreveremos alguns deles a seguir.

Back Pressure

Quando um roteador está congestionado, ele propaga para o roteador em *upstream* uma informação solicitando a redução da taxa de pacotes de saída. Essa ação pode ser recursiva e acontecer (propagar) do roteador até a origem dos dados. Este mecanismo é denominado *back pressure*.

Choke Packet

Um **choke packet (pacote de alerta)** é um pacote enviado da origem ao roteador para informá-lo sobre o congestionamento. Este tipo de controle é similar à mensagem de tráfego (*source-quench*) do ICMP.

Sinalização Implícita

Um *host* de origem pode detectar um sinal de congestionamento implícito e reduzir a taxa de transmissão. Por exemplo, um mero atraso da confirmação do recebimento de um pacote pode ser um sinal de que a rede está congestionada. Veremos este tipo de sinalização quando discutirmos o controle de congestionamento do TCP na Seção 23.4.

Hidden page

O transmissor dispõe de duas partes da informação: a informação sobre o tamanho da janela recebida do receptor e o tamanho da janela de congestionamento. O tamanho real da janela é o menor valor entre os tamanhos dessas duas janelas.

$$\text{Tamanho real da janela} = \min(\text{tamanho da janela do receptor}, \text{tamanho da janela de congestionamento})$$

Evitando o Congestionamento

Para evitar o congestionamento, o TCP transmissor dispõe de duas estratégias: a primeira denominada **SSAI – Slow Start and Additive Increase** e a segunda denominada **MD – Multiplicative Decrease**.

Slow Start Na abertura da conexão, o TCP configura o tamanho da janela de congestionamento para o tamanho máximo de um segmento. Para cada segmento confirmado, o TCP aumenta o tamanho da janela de congestionamento para duas vezes o tamanho máximo de um segmento até atingir o limiar da metade do tamanho admissível da janela. Isto é denominado *slow start*. Nesse processo, o tamanho da janela de congestionamento aumenta exponencialmente. O transmissor envia um segmento, recebe uma confirmação, aumenta o tamanho da janela para dois segmentos, transmite dois segmentos, recebe as confirmações para os dois segmentos, aumenta o tamanho para quatro segmentos, envia quatro segmentos, recebe as confirmações para os quatro segmentos, aumenta o tamanho da janela para oito segmentos e assim por diante. Noutras palavras, após receber a terceira resposta de confirmação, o tamanho da janela aumentou para oito segmentos. A taxa exponencial é ($2^3 = 8$). O processo *slow start* é utilizado juntamente com a estratégia *additive increase*.

Additive Increase Para evitar a ocorrência de congestionamento, o crescimento exponencial da taxa de transmissão deve ser freado. Após o tamanho da janela atingir o limiar, o aumento do tamanho da janela passa a ser um segmento para cada resposta de confirmação, até mesmo se uma resposta de confirmação é dedicada a vários segmentos ao mesmo tempo. A estratégia *additive increase* continua à medida que as respostas de confirmação chegam, antes dos *time-outs* serem disparados ou o tamanho da janela de congestionamento alcançar o valor da janela do receptor.

Multiplicative Decrease Mesmo tomando as medidas preventivas o congestionamento pode ocorrer. Caso ocorra, o tamanho da janela de congestionamento deve ser diminuído. O único modo do transmissor detectar a existência de um congestionamento é através de um segmento perdido. O transmissor assume o congestionamento, se ele não receber uma resposta ACK para um segmento antes do relógio (*timer*) de retransmissão disparar o reenvio do pacote. Como as redes hoje em dia têm uma qualidade de transmissão excelente é mais provável que um segmento seja perdido que corrompido. A estratégia é olhar o relógio de retransmissão. Se ocorrer *time-out*, o limiar da janela deve ser configurado para metade do tamanho da última janela de congestionamento e o tamanho da janela de congestionamento retorna a 1. Assim, o transmissor é forçado a retornar para a fase *slow start*. Observe que o limiar é reduzido para metade da janela de congestionamento atual toda vez que ocorrer um *time-out*. Isto significa que o valor do limiar é reduzido exponencialmente (*multiplicative decrease*). A Figura 23.8 ilustra a idéia.

Controle de Congestionamento no Frame Relay

O congestionamento nas redes Frame Relay diminui o *throughput* e aumenta o atraso. O *throughput* alto e um atraso pequeno são as metas principais do protocolo Frame Relay. Vimos que o Frame Relay não possui mecanismo de controle de fluxo e permite que o usuário transmita rajada de dados. Logo, as redes Frame Relay são potencialmente passíveis ao congestionamento, requerendo, assim, um mecanismo eficaz de controle de congestionamento.

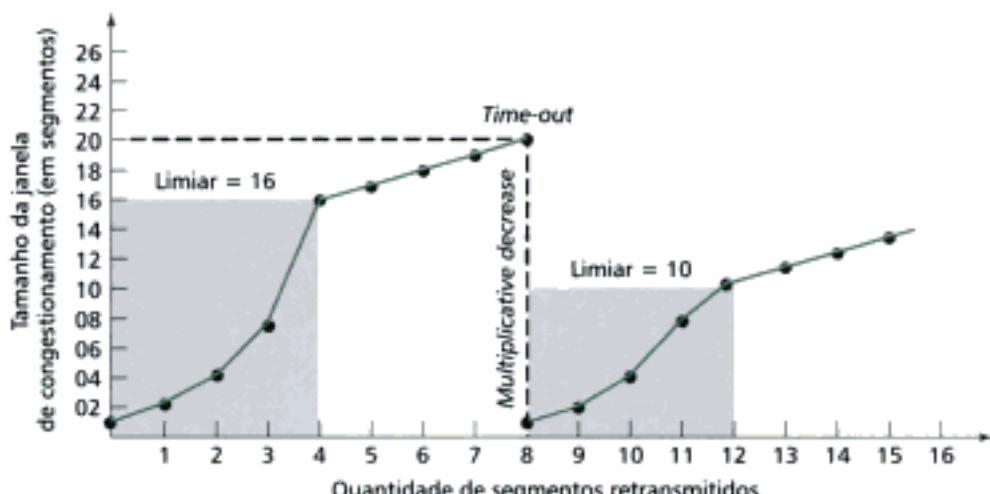


Figura 23.8 Multiplicative decrease.

Evitando o Congestionamento

Para evitar o congestionamento o protocolo Frame Relay possui dois bits no frame para advertir explicitamente os hosts de origem e de destino sobre a existência de congestionamento na rede.

BECN O bit **BECN** (Backward Explicit Congestion Notification) adverte um transmissor sobre a existência de congestionamento na rede. Como isso é possível se os frames viajam na rede se afastando do transmissor? De fato, existem dois métodos: o nó de comutação (comutador) pode utilizar frames de resposta do receptor (modo full-duplex) ou senão o comutador pode usar uma conexão predefinida (DLCI = 1023) para enviar frames especiais para este propósito específico. O transmissor pode responder a isso reduzindo simplesmente a taxa de transmissão de dados. A Figura 23.9 mostra o uso do bit BECN.

FECN O bit **FECN** (Forward Explicit Congestion Notification) adverte um receptor sobre a existência de congestionamento na rede. Pode parecer que o receptor seja passivo, não podendo fazer nada para aliviar o congestionamento. Contudo, o protocolo Frame Relay assume que o transmissor e o receptor estão se comunicando e usando algum tipo de controle de fluxo implementado nas camadas mais altas. Por exemplo, se houver um mecanismo de confirmação nas camadas mais altas, o receptor poderá atrasar a confirmação, forçando assim o transmissor a reduzir a taxa de transmissão. A Figura 23.10 ilustra o uso do FECN.

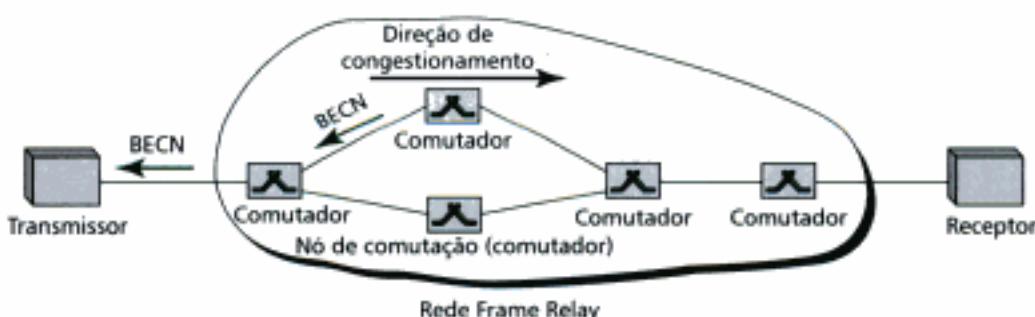


Figura 23.9 BECN.



Figura 23.10 FECN.

Quando as duas extremidades estão se comunicando através de uma rede Frame Relay, podem acontecer quatro situações com relação ao congestionamento. A Figura 23.11 mostra as quatro situações e os valores dos bits FECN e BECN.

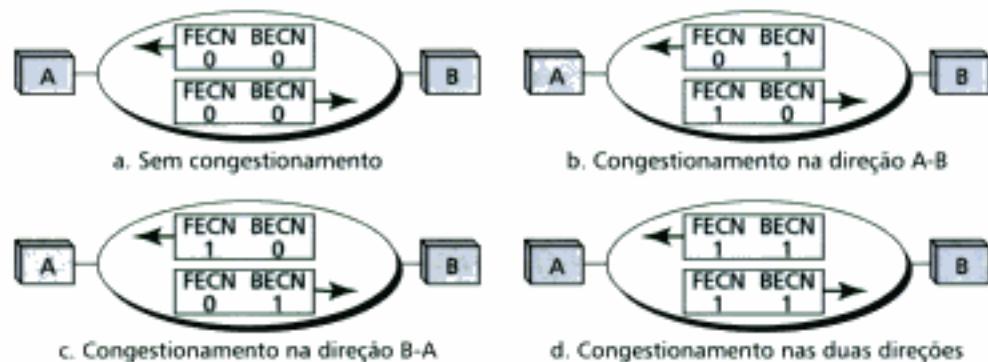


Figura 23.11 Quatro casos de congestionamento.

23.5 QUALIDADE DE SERVIÇO

A expressão **qualidade de serviço (QoS – Quality of Service)** de uma *internetworking* é uma questão muito mais discutida do que definida. Podemos definir informalmente a QoS como uma medida do desempenho de um sistema de transmissão que reflete a sua qualidade de transmissão e a disponibilidade de serviço.

Características do Fluxo

Tradicionalmente, quatro tipos de características são atribuídas a um fluxo de dados: confiabilidade, atraso, jitter e largura de banda, conforme mostra a Figura 23.12.



Figura 23.12 Características do fluxo.

Confiabilidade

Confiabilidade é uma característica necessária ao fluxo. A falta de confiabilidade significa que pacotes de dados serão perdidos, ou então, que as respostas de confirmação estão sendo perdidas, o que leva à retransmissão. Entretanto, a sensibilidade dos processos finais à confiabilidade não é a mesma. Por exemplo, é muito mais importante para as aplicações de correio eletrônico, transferência de arquivos e acesso à Internet ter uma transmissão confiável do que para uma conferência de áudio ou telefônica.

Atraso

O **atraso** de transmissão da origem ao destino é outra característica do fluxo. Outra vez, as aplicações podem tolerar atrasos em diferentes graus. Nesse caso, a telefonia, as conferências de áudio/vídeo e a operação de *login* remoto precisam de atrasos mínimos, enquanto que na transferência de arquivos ou *e-mail* essa característica é menos importante.

Jitter

O **jitter** é a variação no atraso dos pacotes pertencentes a um mesmo fluxo. As aplicações de áudio e vídeo em tempo real não toleram altos valores de *jitter*. Por exemplo, um sistema de *broadcast* de vídeo em tempo real é inútil se houver 2 ms de atraso para o primeiro e o segundo pacotes e um atraso de 60 ms para o terceiro e quarto pacotes. Por outro lado, não importa se os pacotes transportando informação em um arquivo têm atrasos diferentes. A camada de transporte do *host* de destino espera até que todos os pacotes cheguem antes de entregá-los à camada de aplicação.

Largura de Banda

Os diferentes tipos de aplicações precisam de larguras de banda diferentes. Numa conferência de vídeo precisamos enviar milhões de *bits* por segundo para restaurar as cores do monitor enquanto que a quantidade total de *bits* num *e-mail* pode nem mesmo chegar à casa de um milhão.

Classes de Fluxo

Baseado na característica do fluxo, podemos classificar os fluxos em grupos, nos quais cada grupo tem níveis semelhantes. Esta classificação não é formal nem universal; alguns protocolos, como o ATM, têm classes definidas, como veremos adiante.

23.6 TÉCNICAS PARA MELHORAR A QoS

Na Seção 23.5 tentamos definir a QoS em termos de algumas características inerentes a ela. Nesta seção, discutiremos algumas técnicas que podem ser utilizadas para melhorar a QoS. Discutiremos rapidamente os quatro métodos mais comuns: a política de filas, a modelamento do tráfego, controle de admissão e reserva de recursos.

Política de Filas

Pacotes de diferentes fluxos chegam ao *switch* ou roteador para serem processados. Boas políticas de filas tratam os diferentes fluxos de um modo satisfatório e apropriado. Muitas políticas de filas foram desenvolvidas para melhorar a QoS. Estudaremos três delas: a fila FIFO, a fila de prioridade e a WFQ (Weighted Fair Queuing).

Fila FIFO

Na fila **FIFO** (First-in First-out), os pacotes alocados num *buffer* (fila) esperam até que o nó (roteador ou *switch*) possa realmente processá-los. Se a taxa de chegada dos pacotes for maior que a taxa de processamento dos pacotes, a fila será preenchida e os novos pacotes serão descartados. Uma fila FIFO é semelhante a uma fila de banco, a primeira pessoa a entrar na fila geralmente é a primeira a sair dela. A Figura 23.13 ilustra a visão conceitual da fila FIFO.

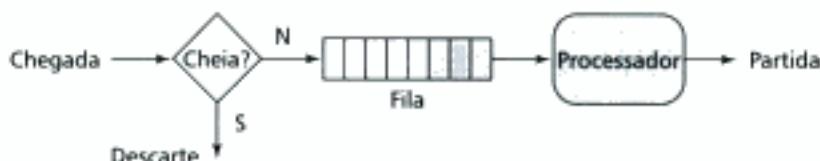


Figura 23.13 Fila FIFO.

Fila de Prioridade

Numa **fila de prioridade** os pacotes recebem primeiramente uma classe de prioridade. Cada classe de prioridade possui uma fila própria. Os pacotes na fila de prioridade mais elevada são processados primeiro. Os pacotes na fila de prioridade mais baixa são processados por último. Observe que o sistema não pára de fazer o uso de uma fila até que ela esteja vazia. A Figura 23.14 mostra a situação para o caso onde existem dois níveis de prioridade (por simplicidade).

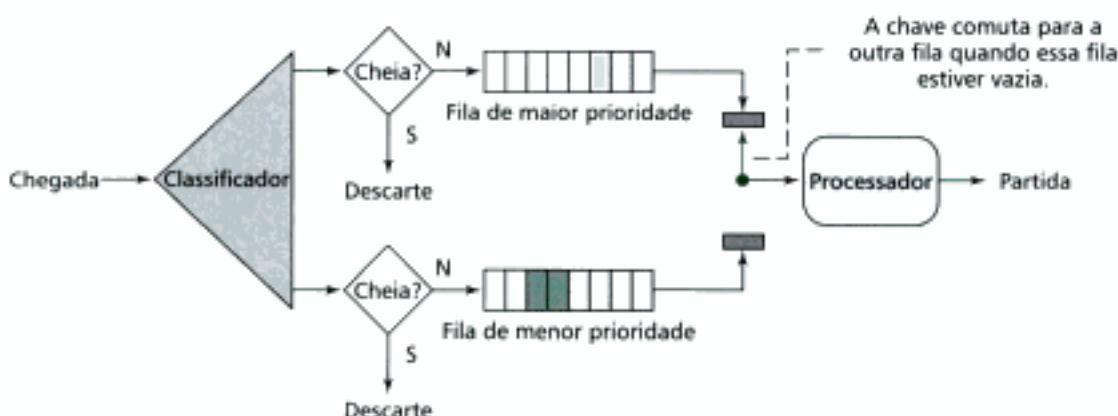


Figura 23.14 Fila de prioridade.

Em geral, as filas de prioridade fornecem melhores QoSs que as filas FIFO visto que o tráfego de alta prioridade, como multimídia, consegue alcançar o destino com menos atrasos. Entretanto, há uma desvantagem potencial. Se houver um fluxo contínuo numa fila de alta prioridade, os pacotes nas filas de prioridade menores nunca terão a chance de serem processados. Esta é uma condição denominada *starvation* (morte por inanição).

Weighted Fair Queuing

Uma política de fila melhor é o **weighted fair queuing**. Nesta técnica, os pacotes ainda são atribuídos a diferentes classes e admitidos em filas diferentes. Porém, as filas recebem pesos baseados na prioridade de cada uma delas; uma prioridade alta significa um peso maior. O sistema processa os pacotes em cada fila no modo de petição, isto é, com a quantidade de pacotes selecionados em cada fila baseada no peso correspondente. Por exemplo, se os pesos são 3, 2 e 1, três pacotes são processados na primeira fila, dois na segunda fila e um na terceira fila. Se o sistema não impuser prioridade sobre as classes, todos os pesos são iguais. Dessa maneira, temos uma fila de distribuição com prioridade. A Figura 23.16 mostra a técnica para três classes.

Modelamento do Tráfego

Modelamento do tráfego é um mecanismo para controlar a quantidade e a taxa do tráfego de dados enviados para uma rede. Duas técnicas comuns conformam o tráfego: *leaky bucket* (balde fúrido) e *token bucket* (balde de fichas).

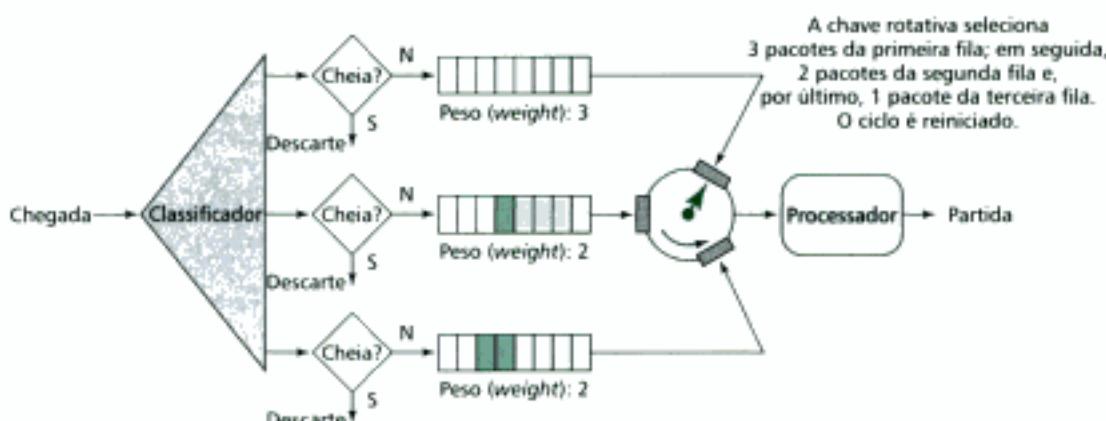


Figura 23.15 Weighted Fair Queuing (WFQ).

Leaky Bucket

Se tomarmos um balde com um pequeno furo no fundo, o vazamento de água do balde acontece a uma taxa constante enquanto houver água no balde. A vazão não depende da taxa com que a água é colocada no balde, exceto quando o balde estiver vazio. A taxa de entrada de água pode variar, mas a taxa de saída permanece constante. Similarmente, numa rede, uma técnica denominada **leaky bucket** pode suavizar um tráfego em rajadas. Parte da rajada é armazenada no *buffer* enquanto a outra parte é transmitida a uma taxa constante. A Figura 23.6 ilustra um *leaky bucket* e os efeitos decorrentes da utilização dessa técnica.

Na figura, assumimos que a rede alocou uma largura de banda de 3 Mbps a um *host*. O uso do *leaky bucket* modela o tráfego de entrada para conformá-lo ao que foi alocado. Na figura, o *host* envia uma rajada de dados a uma taxa de 12 Mbps, durante 2 s, transmitindo um total de 24 megabits de dados. O *host* silencia-se por 5 s e então volta a disparar outra rajada, agora de 2 Mbps por 3 s, transmitindo um total de 6 megabits de dados. Ao todo, o *host* transmitiu 30 megabits de dados em 10 s. Sem o *leaky bucket* a rajada inicial poderia comprometer a rede consumindo mais largura de banda do que foi alocado para este *host*. Portanto, é fácil ver que um *leaky bucket* bem dimensionado consegue evitar um congestionamento. Uma analogia seria considerar uma via de trânsito durante a hora do *rush* (rajada de tráfego). Se todos os motoristas que passam por essa via pudessem escalar os horários para que o fluxo pela via acontecesse de modo distribuído, o congestionamento seria evitado.

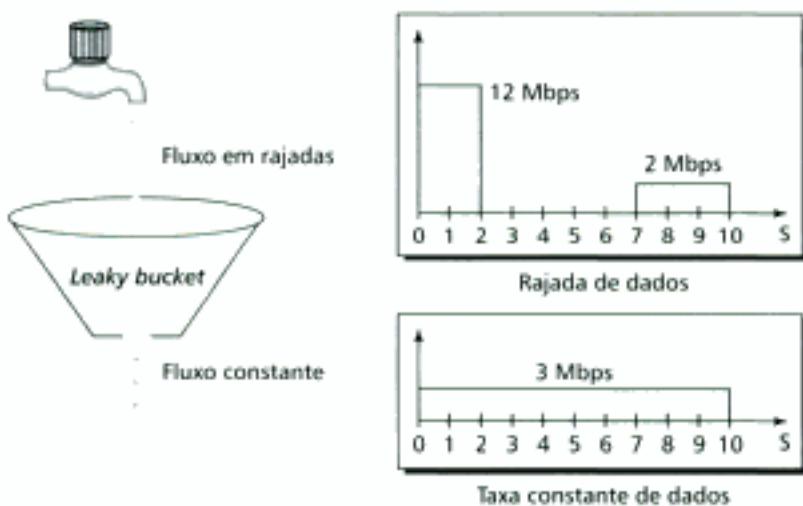


Figura 23.16 Leaky bucket.

Uma implementação simples do *leaky bucket* é mostrada na Figura 23.17. Uma fila FIFO recebe os pacotes. Se o tráfego é formado de pacotes de tamanho fixo (por exemplo, células numa rede ATM), o processo remove uma quantidade fixa de pacotes da fila (a cada ciclo de *clock*). Se o tráfego é formado de pacotes de tamanho variável, o valor da taxa de saída deve ser baseado na quantidade de *bytes* ou *bits* consumidos.

Abaixo vemos os passos de um algoritmo para pacotes de tamanho variável:

1. A cada n ciclos de *clock*, um contador associado ao *buffer* é inicializado.
2. Se n for maior que o tamanho do pacote, transmita o pacote e decremente do contador o tamanho do pacote. Repita este passo até que n seja menor que o tamanho do pacote.
3. Resete o contador e retorne ao passo 1.

O algoritmo *leaky bucket* conforma o tráfego em rajada de dados em um tráfego a uma taxa constante de dados, originando uma taxa de dados média. Este algoritmo pode perder os pacotes se o bucket estiver cheio.

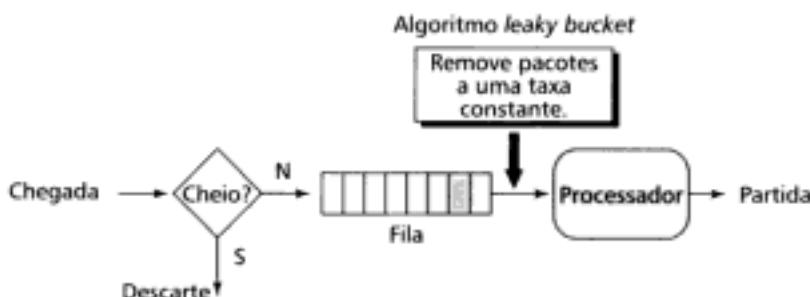


Figura 23.17 Implementação *leaky bucket*.

Token Bucket

O *leaky bucket* é muito restritivo. Ele não dá crédito a um *host* inativo. Por exemplo, se um *host* não estiver enviando dados durante um período, o *buffer* pode ser esvaziado. Já se o *host* tiver disparando uma rajada de dados a técnica *leaky bucket* permite somente uma taxa média. O tempo em que o *host* ficou inativo não é levado em consideração. Por outro lado, o algoritmo **token bucket** permite aos *hosts* inativos "acumularem crédito", na forma de fichas, para uso futuro. Para cada ciclo de *clock*, o sistema envia n fichas ao *buffer*. O sistema remove um *token* para cada célula ou *byte* de dados enviado. Por exemplo, se $n = 100$ e o *host* ficou inativo durante 100 ciclos de *clock*, o *buffer* acumula 10.000 fichas. Nesse caso, o *host* pode consumir todas essas fichas (*tokens*) em um único ciclo com 10.000 células ou o *host* pode levar 1000 ciclos para 10 células por ciclo. Outras palavras, o *host* pode enviar uma rajada de dados uma vez que o *buffer* não está vazio. A Figura 23.18 ilustra a idéia.

O *token bucket* pode facilmente ser implementado através de um contador. O *token* é inicializado com zero. Toda vez que um *token* é adicionado o contador associado ao *buffer* (*token counter*) é incrementado de 1. A cada instante uma unidade de dados é transmitida e o contador é decrementado de 1. Quando o contador chega em zero, o *host* não pode mais transmitir dados.

O algoritmo *token bucket* permite um tráfego de rajadas a uma taxa máxima regulada.

Combinando o Token Bucket e o Leaky Bucket

As duas técnicas podem ser combinadas para dar crédito a um *host* inativo e, ao mesmo tempo, regular o tráfego. O *leaky bucket* é aplicado após o *token bucket*. Além disso, a taxa do *leaky bucket* precisa ser superior à taxa de *tokens* retirados do *buffer*.

Hidden page

Sinalização

O leitor deve se lembrar que o IP é um protocolo sem conexão, baseado em datagramas e utiliza uma rede de comutação de pacotes. Podemos implementar um modelo de fluxo baseado em um protocolo não orientado à conexão? A solução é um protocolo de sinalização para rodar sobre o IP que proporcione mecanismos de sinalização que tome conta das reservas de recursos. Este protocolo é denominado **Resource Reservation Protocol (RSVP)** que será analisado em breve.

Especificações do Fluxo

Quando uma fonte de tráfego faz uma reserva é necessário que ela especifique o fluxo. Uma especificação de fluxo é composta de duas partes: Rspec (especificação dos recursos) e Tspec (especificação do tráfego). A Rspec define os recursos que o fluxo precisa reservar (*buffer*, largura de banda, etc.). A Tspec é a caracterização do tráfego em si.

Admissão

Após o roteador receber a especificação do fluxo de uma determinada aplicação deve decidir entre aceitar ou negar o serviço. A decisão geralmente está baseada no comprometimento que o roteador faz com os fluxos já iniciados e a disponibilidade atual de recursos.

Classes de Serviços

Duas classes de serviços foram definidas para atender ao Serviços Integrados: serviço garantido e o serviço de carga controlada.

Classe Serviço Garantido

Este tipo de serviço foi desenvolvido para tratar o tráfego de aplicações em tempo real, assegurando-lhes largura de banda, um limite rígido de atraso fim a fim e uma proteção contra a perda de pacotes nas filas, para os pacotes que estiverem obedecendo o perfil de tráfego contratado. O atraso fim a fim é a soma dos atrasos nos roteadores, mais o atraso de propagação médio e o atraso dos mecanismos de *setup*. Somente a primeira parcela, a soma dos atrasos provocados pelos roteadores, pode ser garantido pelo roteador. Este tipo de serviço garante que os pacotes chegam dentro do tempo de entrega e não serão descartados se o fluxo estiver dentro da especificação do Tspec. Podemos afirmar que os serviços garantidos são de fato serviços quantitativos, aonde o atraso fim a fim e a taxa de dados devem ser definidos pela aplicação.

Classe Serviço de Carga Controlada

Este tipo de serviço foi desenvolvido para as aplicações que admitem algum tipo de atraso, mas são muito sensíveis à sobrecarga na rede e à perda de pacotes. Bons exemplos destes tipos de aplicações são a transferência de arquivos (*download/upload*), *e-mail* e o acesso à Internet. O serviço de carga controlada é um tipo de serviço qualitativo no qual a aplicação define a possibilidade de pouca ou nenhuma perda de pacotes.

RSVP

Na abordagem de Serviços Integrados uma aplicação precisa reservar os recursos necessários. Na discussão dos IntServ, a reserva de recursos é para estabelecer um fluxo. Isto significa que, se pretendemos usar o IntServ na camada IP, precisamos estabelecer um fluxo, um tipo de circuito virtual, sem vínculo com o IP, o qual foi originalmente desenvolvido para as redes de comutação de pacotes. Um circuito virtual necessita de um sistema de sinalização para configurá-lo antes do início do tráfego de dados. O Resource Reservation Protocol (RSVP) é um protocolo de sinalização que auxilia o IP na tarefa de criar e, consequentemente, fazer a reserva dos recursos. Antes de discutirmos o RSVP é importante enfatizarmos que ele é um protocolo independente, isto é, separado do modelo Serviços Integrados. Logo, no futuro, é provável que o encontremos noutras abordagens da QoS.

Árvores Multicast

O RSVP é diferente de outros sistemas de sinalização que estudamos. Comentamos antes que o RSVP é um sistema de sinalização desenvolvido para aplicações em *multicasting*. Contudo, o RSVP também pode ser utilizado em aplicações *unicasting* porque o *unicast* é um caso especial de *multicast* com um único membro no grupo. A razão é simples, o RSVP assim implementado é capaz de reservar recursos para todos os tipos de tráfego, incluindo multimídia, que freqüentemente usa *multicasting*.

Reserva Baseada no Receptor

No RSVP são os receptores e não os transmissores que fazem a reserva. Esta estratégia assemelha-se com a proposta de outros protocolos de *multicasting*. Por exemplo, nos protocolos de roteamento *multicast* são os receptores que tomam a decisão de se juntar ou deixar um grupo *multicast*.

Mensagens RSVP

O RSVP possui muitos tipos de mensagens. Entretanto, para os nossos propósitos, discutiremos somente duas delas: **PATH** e **RESV**.

Mensagens PATH Lembre-se que no protocolo RSVP são os receptores do fluxo que fazem as reservas. Entretanto, antes da concretização das reservas, os receptores não sabem predizer o caminho percorrido pelos pacotes. A informação de caminho é fundamental para o estabelecimento da reserva. Para resolver o problema, o RSVP usa as mensagens **PATH**. Uma mensagem PATH viaja do transmissor para todos os receptores no caminho (*multicast*). Desse modo, essa mensagem armazena e transporta a informação necessária aos receptores. Uma nova mensagem é criada quando há divergência de caminho. A Figura 23.19 ilustra as mensagens PATH.

Mensagens RESV Após receber a mensagem PATH, o receptor transmite uma mensagem RESV. A mensagem RESV viaja na direção do transmissor (*upstream*) e faz a reserva dos recursos nos roteadores que suportam RSVP. No caminho, se um roteador não suportar o protocolo RSVP, a decisão de roteamento dos pacotes será baseada no serviço de melhor esforço discutido anteriormente. A Figura 23.20 ilustra as mensagens RESV.

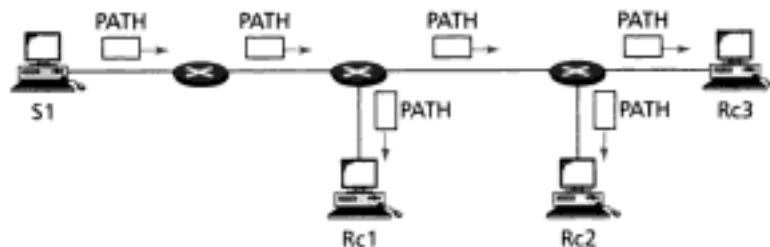


Figura 23.19 Mensagens PATH.

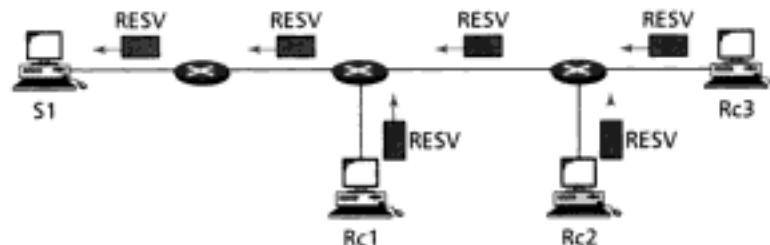


Figura 23.20 Mensagens RESV.

Hidden page

Problemas com os Serviços Integrados

Há pelo menos dois problemas com os Serviços Integrados que podem retardar, senão evitar, a implementação completa deles na Internet: escalabilidade e tipos de serviços limitados.

Escalabilidade

A abordagem de Serviços Integrados exige que cada roteador mantenha a informação para cada fluxo individualmente. Atualmente, como a Internet já é um monstro, isto constitui um problema sério.

Tipos de Serviços Limitados

Os Serviços Integrados proporcionam somente dois tipos de serviços: garantido e controle de carga. Os críticos à abordagem IntServ argumentam que as aplicações futuras podem necessitar de outros tipos de serviços.

23.8 SERVIÇOS DIFERENCIADOS

Os **Serviços Diferenciados (DS ou Diffserv)** foram introduzidos pela IETF (Internet Engineering Task Force) para contornar as deficiências dos Serviços Integrados. Duas mudanças fundamentais foram realizadas:

1. O processamento principal foi movido do núcleo para a borda da rede. Isto resolve o problema da escalabilidade. Os roteadores não precisam armazenar informações sobre os fluxos. As aplicações ou *hosts* definem o tipo de serviço de que precisam toda vez que enviam um pacote.
2. O serviço de fluxo foi substituído pela classe de serviço. O roteador roteia os pacotes baseado na classe de serviço definida no pacote e não no fluxo. Isto resolve o problema da limitação dos tipos de serviço.

Serviços Diferenciados é uma abordagem QoS baseada nas classes de serviços, desenvolvida para o protocolo IP.

Campo DS

No Diffserv, cada pacote contém um campo denominado DS. O valor desse campo é configurado na borda da rede por um *host* ou pelo primeiro roteador designado como roteador de borda. O IETF propôs substituir o campo ToS (Type of Service) existente na versão IPv4 ou o campo de classe na versão IPv6 pelo campo DS, conforme ilustra a Figura 23.23.

O campo DS possui dois sub-campos: DSCP e CU. O DSCP (Differentiated Services Code Point) é um sub-campo de 6-bits que define o **Per-Hop Behavior (PHB)**. O sub-campo CU (Currently Unused) não é utilizado atualmente.

O Diffserv possibilita que o nó (roteador) utilize o sub-campo DSCP de 6-bits como um índice em uma tabela definindo o mecanismo de controle de pacotes para o pacote que estiver sendo processado.

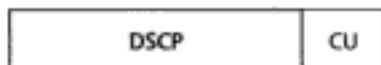


Figura 23.23 Campo DS.

Hidden page

23.9 QoS EM REDES COMUTADAS

Discutimos as propostas de abordagem QoS no protocolo IP (rede de comutação de pacotes). Vamos discutir agora abordagens QoS usadas em duas redes comutadas: Frame Relay e ATM. As duas são redes de comutação de circuitos virtuais que necessitam de um protocolo de sinalização como o RSVP.

QoS nas Redes Frame Relay

Quatro atributos diferentes para controle do tráfego foram planejados para as redes Frame Relay: taxa de acesso, tamanho de rajada comprometido (B_c), taxa de informação comprometida (CIR) e tamanho excedente da rajada (B_e). Estes atributos são configurados durante a negociação entre o usuário e a rede. Nas conexões PVC elas são negociadas uma única vez. Nas conexões SVC elas são negociadas em cada conexão durante o *setup*. A Figura 23.25 apresenta a relação entre estas quatro formas de medida.

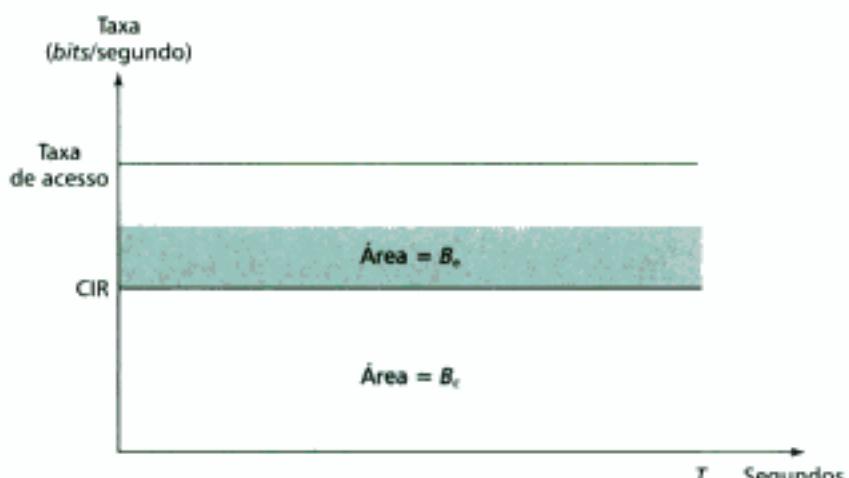


Figura 23.25 Relacionamento entre os atributos de controle de tráfego.

Taxa de Acesso

Para cada conexão é definida uma **taxa de acesso** em *bits* por segundo. A taxa de acesso atualmente depende da largura de banda do canal que conecta o usuário à rede. O usuário nunca excede essa taxa. Por exemplo, se um usuário estiver conectado a uma rede Frame Relay através de uma linha T-1, a taxa de acesso é 1,544 Mbps e nunca excederá esse valor.

Tamanho de Rajada Comprometido

Para cada conexão, o Frame Relay define um **tamanho de rajada comprometido B_c** . Esta é a quantidade máxima de *bits*, em um intervalo de tempo predeterminado, que uma rede está comprometida a aceitar sem descartar *frames* ou configurar o bit DE. Por exemplo, se a B_c garantida é 400 kbits por um período de 4s, o usuário poderá enviar os 400 kbits nesse intervalo de 4 s sem se preocupar com a perda de *frames*. Observe que esta taxa não é definida por segundo. Ela é uma medida cumulativa. O usuário pode enviar 300 kbits durante o primeiro segundo, não transmitir nada no segundo e no terceiro segundos e finalmente enviar os 100 kbits restantes no último segundo.

CIR

A **taxa de informação comprometida (CIR)** é similar, conceitualmente falando, ao tamanho de rajada comprometido, exceto que a CIR define uma taxa média em *bits* por segundo. Se o usuário seguir esta taxa continuamente, a rede compromete-se em entregar todos os *frames*. Entretanto, como essa medida é especificada em função da média, o usuário pode transmitir dados numa taxa mais elevada que a CIR, durante um tempo, e mais baixa em um outro momento. Uma vez que a média durante um período de tempo predefinido é mantida, os *frames* serão entregues.

A quantidade cumulativa de *bits* enviado durante um período predeterminado não pode exceder a B_c . Observe que a CIR não é uma medida independente. Ela pode ser calculada a partir da seguinte fórmula:

$$\text{CIR} = \frac{B_c}{T} \text{ (em bps)}$$

Por exemplo, se a B_c for 5 kbytes por 5s, a CIR será 5000/5 ou 1 kbps.

Tamanho Excedente da Rajada

Para cada conexão, o Frame Relay define um **tamanho excedente da rajada (B_e)**. Esta é a quantidade máxima de *bits* excedendo a B_c que um usuário pode enviar durante um período de tempo predeterminado. A rede permanece comprometida em transferir esses *bits* se não houver congestionamento. Observe que, nesse caso, há menos comprometimento que no caso do B_c . A rede compromete-se condicionalmente.

Taxa do Usuário

A Figura 23.26 ilustra rajadas de dados transmitidas por um usuário. Se o usuário nunca excede a B_c , a rede compromete-se em transmitir todos os *frames* sem descartar nenhum deles. Se o usuário excede B_c de modo que a quantidade total de *bits* ainda é menor que $B_c + B_e$, a rede compromete-se a transmitir todos os *frames*, caso não haja congestionamento. Se houver, alguns *frames* serão descartados. O primeiro nó de comutação (comutador) que receber os *frames* do usuário possui um contador e configura o bit DE = 1 para os *frames* que excederem B_c . Assim, se ainda houver congestionamento, os comutadores restantes descartam esses *frames*. Observe que o usuário precisa transmitir dados muito rapidamente para exceder o nível B_c . Uma vez que o nível de transmissão não excede $B_c + B_e$, existe uma chance de todos os *frames* chegarem ao destino sem sofrer descartes. Porém, no momento que o usuário excede o nível $B_c + B_e$, todos os *frames* enviados após o instante onde ocorreu o excesso serão descartados pelo primeiro comutador.

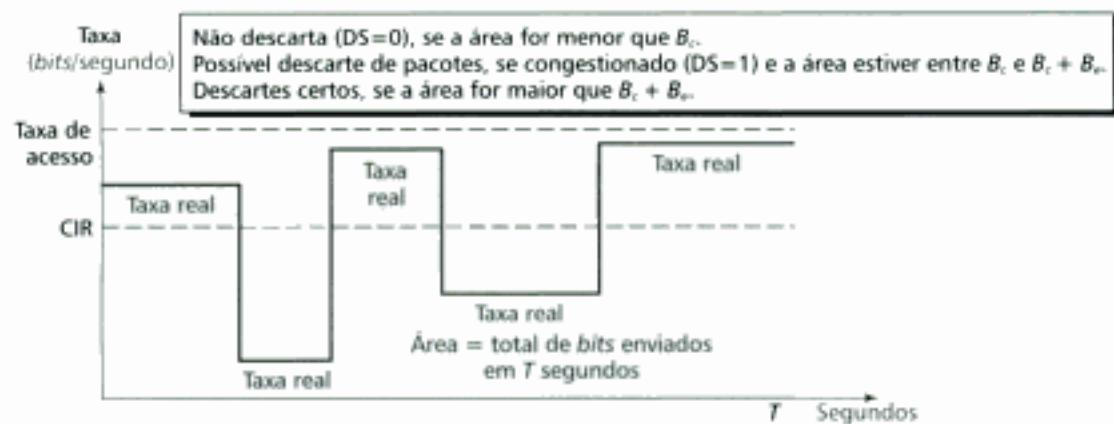


Figura 23.26 Taxa de usuário em relação a B_c e $B_c + B_e$

QoS nas Redes ATM

O QoS nas redes ATM baseia-se nas classes, nos atributos relacionados ao usuário e nos atributos relacionados à rede.

Hidden page

Hidden page

Hidden page

Questões de Múltipla Escolha

17. A _____ é definida como a quantidade de *bits* enviados dividida pelo intervalo de tempo decorrido durante a transmissão.
- Taxa média de dados
 - Tamanho máximo da rajada
 - Largura de banda efetiva
 - Taxa constante de *bits*
18. A _____ é a taxa de pico de um tráfego.
- Taxa média de dados
 - Taxa máxima de dados
 - Tamanho máximo da rajada
 - Largura de banda efetiva
19. A largura de banda efetiva está baseada na(s) _____.
- Taxa média de dados
 - Taxa máxima de dados
 - Tamanho máximo da rajada
 - Todas as alternativas anteriores
20. Um tráfego em _____ é caracterizado por uma mudança em um intervalo de tempo curto da taxa de transmissão.
- Taxa constante de *bits*
 - Taxa variável de *bits*
 - Rajada
 - Taxa máxima de *bits*
21. Quando a carga é maior que a capacidade, o atraso _____.
- Diminui
 - Aumenta linearmente
 - Tende a infinito
 - Tende a zero
22. _____ é um mecanismo de controle de congestionamento em malha fechada que alivia o congestionamento em um rede.
- Choke packet* (pacote de alerta)
 - Sinalização explícita
 - Sinalização implícita
 - Todas as alternativas anteriores
23. Para um sistema usando TCP, o tamanho da janela de transmissão é determinado pelo _____.
- Receptor
 - Transmissor
 - Congestionamento
 - (a) e (c)
24. *Slow start* é utilizado em conjunto com _____ como uma estratégia TCP para controle de congestionamento.
- Additive increase*
 - Additive decrease*
 - Multiplicative increase*
 - Multiplicative decrease*
25. O FECN informa _____ enquanto que o BECN informa _____ sobre o congestionamento.
- O host de destino; a interface
 - O host de destino; o transmissor
 - O transmissor, o host de destino
 - A interface; o transmissor
26. _____ é uma das características do fluxo na qual o atraso varia para os pacotes pertencentes ao mesmo fluxo.
- Choke packet*
 - Throughput*
 - Additive increase*
 - Jitter*
27. Numa fila _____ o primeiro pacote a chegar será o primeiro pacote a sair da fila.
- FIFO
 - LIFO
 - De prioridade
 - WFQ
28. O método de modelamento _____ dá créditos a um host durante o tempo inativo.
- Leaky bucket*
 - Token bucket*
 - Traffic bucket*
 - Bursty bucket*
29. _____ é uma abordagem QoS baseada no fluxo, desenvolvida para o protocolo IP.
- Serviços Integrados
 - Serviços Diferenciados
 - RSVP
 - Árvores de *multicast*
30. Um protocolo de sinalização que auxilia o IP a criar um fluxo é denominado _____.
- Serviços Integrados
 - Serviços Diferenciados

Hidden page

- c. 70.000
d. 5.000
46. A _____ é a fração das células entregues com erros.
 a. CLR
 b. CTD
 c. CDV
 d. CER
47. Se a CTD mínima vale $10\mu s$ e a CTD máxima vale $1\mu s$, a _____ vale $9\mu s$.
 a. CLR
 b. CTD
 c. CDV
 d. CER

Exercícios

48. O campo endereços de um frame *Frame Relay* é 1011000100010110. Há congestionamento na direção direta? E na direção reversa?
49. Um frame vai de A para B. Há congestionamento nas duas direções. O bit FECN é configurado? E o bit BECN?
50. Em um *leaky bucket* (balde furado), qual deve ser a capacidade do balde se a taxa de saída é 4 litros/min e ocorre uma rajada de entrada de 100 litros/min, durante 12s, e não mais nenhuma entrada por 48 s?
51. Uma interface de saída de um *switch* foi projetada para usar o algoritmo *leaky bucket* para enviar 8.000 bytes/s (ciclo). Se os seguintes frames são recebidos na seqüência, apresente os frames que são enviados durante cada segundo.
Frames 1, 2, 3, 4: 4.000 bytes cada
Frames 5, 6, 7: 3200 bytes cada
Frames 8, 9: 400 bytes cada
Frames 10, 11, 12: 2.000 bytes cada
52. Um usuário está conectado a uma rede *Frame Relay* através de um linha T-1. O CIR contratado é 1Mbps, com um B_c de 5 milhões por 5s e B_r de 1 milhão por 5s. Responda às seguintes questões:
 a. Qual é a taxa de acesso?
 b. Um usuário pode transmitir dados a 1,6Mbps?
- c. Um usuário pode transmitir dados constantemente a 1Mbps? Nesse caso, é assegurado que os frames nunca serão descartados?
- d. Um usuário pode transmitir dados constantemente a 1,2 Mbps? Nesse caso, é assegurado que os frames nunca serão descartados? Se a resposta for não, é assegurado que os frames serão descartados somente se houver congestionamento?
- e. Repita item (d) para uma taxa constante de 1,4 Mbps.
- f. Qual é a taxa máxima de dados que o usuário pode fazer uso sem se preocupar com o descarte de frames?
- g. Se o usuário deseja correr o risco, qual é a taxa máxima de dados que pode ser utilizada sem a chance de ocorrer descartes, supondo que não há congestionamento?
53. No Exercício 53, o usuário envia dados a 1,4 Mbps por 2s e nada nos próximos 3s. Se não houver congestionamento, existe a possibilidade de ocorrência de descartes? E se houver congestionamento?
54. Se cada célula precisa de $10\mu s$ para alcançar o destino, qual é a CTD?
55. Uma rede perdeu 5 células em 10.000 transmitidas e 2 chegaram com erros. Qual é a CLR? Qual é a CER?

PARTES VI

CAMADA DE APLICAÇÃO

Esta parte do livro explora as características de muitos programas aplicativos disponíveis para a camada mais alta, camada cinco, do modelo da Internet. A camada de aplicação permite que pessoas comuns, sem nenhum conhecimento de redes, acessem a Internet. Podemos dizer que as outras quatro camadas foram criadas para que as pessoas possam utilizar programas aplicativos.

A Figura 1 ilustra a posição da camada de aplicação no modelo de cinco camadas da Internet. Observe que a camada de aplicação é última no modelo. Acima dela estão os usuários e abaixo dela está a camada de transporte. Isto significa que a camada de aplicação recebe serviços da camada de transporte e oferece serviços aos usuários.

A camada de aplicação possibilita que um usuário, seja ele um ser humano ou um *software*, acesse a rede. Ela disponibiliza interfaces e suporte a serviços como *e-mail*, acesso e transferência de arquivos remotamente, acesso a World Wide Web (WWW) e muitos outros.



Figura 1 Posição da camada de aplicação.

Finalidades da Camada de Aplicação

Podemos dizer que há três tipos gerais de questões relacionadas a esta camada: o paradigma cliente-servidor, endereçamento e tipos de serviços.

Paradigma Cliente-Servidor

Os programas da camada de aplicação estão baseados no conceito de clientes e servidores. O propósito de uma rede e, em particular da Internet global, é oferecer serviços aos usuários. Um usuário localizado em um ponto qualquer do globo deseja receber serviços de um computador localizado noutro ponto remoto. Por exemplo, um usuário deseja fazer um *download* de um computador remoto. Nesse caso, ambos computadores devem rodar programas. O computador local roda um programa que solicita os serviços de outro programa localizado em um computador remoto. Discutiremos o paradigma cliente-servidor no Capítulo 24.

Endereçamento

Um cliente e um servidor se comunicam usando algum esquema de endereçamento. Quando um cliente solicita um serviço de um servidor é necessário que inclua o endereço do servidor como endereço de destino, assim como o seu próprio endereço com o endereço de origem. O endereço de origem é necessário para que o servidor saiba para onde enviar a resposta. Quando o servidor responde a uma solicitação, ele inverte a posição dos endereços; o endereço próprio passa a ser o endereço de origem e o endereço do cliente assume a posição de endereço de destino.

Entretanto, o mecanismo de endereçamento em um aplicação não é como um endereço para as outras camadas; cada aplicação tem um formato próprio de endereço. Por exemplo, um endereço de *e-mail* pode ser *forouzan@fhda.edu* enquanto que um endereço para acessar uma página da Web pode ser *http://www.fhda.edu*.

Podemos dizer que parte do endereço está relacionado ao endereço de porta do servidor e a estrutura do diretório onde o programa servidor está localizado. Entretanto, a parte principal é o alias (nome alternativo) para o endereço do *host* remoto. O programa aplicativo usa um nome em vez do endereço IP. Embora este tipo de endereço seja muito conveniente de se lembrar e utilizar pelas pessoas, não é conveniente ao protocolo IP quando é aberta a comunicação com o servidor. Um endereço alias é "amigável", mas deve ser mapeado em um endereço IP. Uma aplicação precisa dos serviços de outra aplicação para mapear o endereço alias em endereço IP. Esta aplicação é denominada DNS. O DNS não é utilizado explicitamente pelos usuários; mas sim por outros programas para realizar o mapeamento. Discutiremos o DNS no Capítulo 25.

Tipos de Serviços

A camada de aplicação foi desenvolvida para oferecer serviços diferentes aos usuários (pessoa ou outro programa). O serviço mais comum (SMTP) permite que um usuário envie mensagens para outro usuário na Internet. Este serviço é o famoso correio eletrônico (*e-mail*) e guarda muitas semelhanças com o serviço de correio tradicional. Outro serviço comum é a transferência de arquivo. Um usuário pode transferir um arquivo do seu computador para um servidor (*upload*) ou transferir do servidor para o seu computador (*download*). Esta aplicação é denominada FTP. Estudaremos estes serviços no Capítulo 26.

A invenção da World Wide Web anunciou uma nova era para a humanidade. Ela trouxe os usuários comuns para dentro da Internet. A WWW é um repositório de informação que o usuário da Internet pode acessar. Para usar a WWW necessitamos chamar um protocolo simples denominado HTTP. Estudaremos a WWW e o HTTP no Capítulo 27.

Recentemente, multimídia na forma de dados de áudio e vídeo atraiu a atenção dos usuários da Internet. Eles passaram a ouvir músicas armazenadas em um servidor. Eles podem ouvir rádio ou assistir TV através da Internet. Ainda, os usuários podem conversar uns com os outros ou criar um ambiente de teleconferência. Este tipo de serviço é novo e está crescendo. Ele trouxe novos conceitos à Internet. A qualidade de serviços (QoS) discutida no capítulo anterior desempenha um papel importante quando estamos utilizando multimídia na rede. O Capítulo 28 é totalmente dedicado a esta questão.

Suporte

Para ser útil aos usuários, uma aplicação deve ser suportada pelos serviços oferecidos pela camada mais baixa (camada de transporte). O tipo de suporte necessário é diferente para aplicações diferentes. Classificamos este suporte em três categorias: confiabilidade, *throughput* e atraso.

Confiabilidade

Algumas aplicações dependem pesadamente da confiabilidade. Entre elas estão o *e-mail* e a transferência de arquivos. Quando estamos lendo um *e-mail* não gostamos quando vemos que ele foi corrompido ou que, durante o andamento, um *download* foi perdido ou interrompido. Estes tipos de aplicação precisam adicionar confiabilidade como parte do protocolo de aplicação ou usar os serviços de confiabilidade de um protocolo da camada de transporte (como o TCP). Outras aplicações não são sensíveis à confiabilidade. Se uma pequena parte de uma música baixada da Internet é perdida, pode ser que o usuário nem perceba.

Throughput

O *throughput* máximo, a quantidade máxima de dados que podem ser transferidos numa certa unidade de tempo, é um critério requerido por algumas aplicações. Em geral, as aplicações de multimídia precisam de *throughput* elevado para serem efetivas. Veremos no Capítulo 28 que transferir arquivos de vídeo ao vivo envolve a transferência de milhões de *bits* durante um curto intervalo de tempo (mesmo se os dados são comprimidos). O *throughput* está intrinsecamente relacionado à largura de banda. Um aplicação que requer um *throughput* elevado também requer uma largura de banda elevada.

Atraso

Algumas aplicações são muito sensíveis ao atraso. Um programa interativo em tempo real não pode tolerar atrasos. Não desejaremos usar a Internet com um serviço de telefonia se, durante a conversação, persistirem longos atrasos. Por outro lado, algumas aplicações não são sensíveis ao atraso. Por exemplo, um *e-mail* pode esperar segundos ou até mesmo horas antes de ser entregue.

Capítulos

A parte seis do livro cobre cinco capítulos. O Capítulo 24 apresenta os conceitos do paradigma cliente-servidor e da interface *socket*. O Capítulo 25 é dedicado ao DNS, um protocolo que mapeia os endereços da camada de aplicação nos endereços da camada de rede. O Capítulo 26 explora dois protocolos comuns, SMTP e FTP, que permitem aos usuários transferir mensagens e arquivos através da Internet. O Capítulo 27 discute os aspectos essenciais da WWW e o protocolo que permite acessá-la, o HTTP. Finalmente, o Capítulo 28 é uma introdução ao uso de multimídia na Internet, uma questão que está evoluindo rapidamente.

Arquitetura Cliente-Servidor: a Interface Socket

Antes de iniciarmos a discussão dos protocolos da camada de aplicação, precisamos compreender a natureza dos programas aplicativos. Precisamos saber que a Internet é baseada na arquitetura cliente-servidor. Para realizar uma tarefa, deve haver um cliente e um servidor. Na Seção 24.1, discutimos a arquitetura. Embora existam muitas maneiras de permitir que um cliente e um servidor se comuniquem, a mais comum é através da interface *socket*, discutida na Seção 24.2. Este capítulo não foi escrito para ensinar programação cliente-servidor, mas para introduzir a idéia através de fluxogramas, uma das ferramentas de programação. Isto fornece os conceitos básicos ao leitor sobre programação cliente-servidor sem aprofundar nos detalhes. Para aqueles que estiverem interessados em exemplos de programas cliente-servidor, veja o Apêndice H.

24.1 ARQUITETURA CLIENTE-SERVIDOR

Há muitas formas de um computador solicitar os serviços de outro computador. De longe, a mais comum delas é **arquitetura cliente-servidor**.

Relacionamento

O propósito de uma rede ou uma *internetwork* é oferecer serviços para os usuários. Um usuário num determinado local deseja receber algum tipo de serviço de um computador remoto. Há apenas um modo de um computador realizar o trabalho; ele deve rodar um programa. Um computador roda um programa para solicitar um serviço de outro computador ou oferecer serviço a outro computador. Isto significa que dois computadores, conectados através de uma internet, devem rodar programas, um para oferecer um serviço e outro para solicitar um serviço.

Deve ficar claro que para usar os serviços disponíveis numa internet é necessário rodar **programas aplicativos** nos dois computadores que estiverem se comunicando. Dito de outra forma, numa internet, os programas aplicativos são as entidades que se comunicam e não os usuários ou os computadores.

À primeira vista parece simples fazer comunicar duas aplicações, uma rodando na máquina local e outra rodando na máquina remota. Entretanto, surgem muitas questões quando queremos abordar a implementação. Algumas das questões mais freqüentes são:

1. Ambas aplicações devem ser capazes de solicitar e receber serviços ou cada qual deve ter sua própria especificidade? Uma solução é ter um programa aplicativo, denominado

cliente, rodando na máquina local solicitando um serviço de outro programa aplicativo, denominado *servidor*, rodando na máquina remota (veja a Figura 24.1).

- Uma aplicação deve oferecer serviços somente a uma aplicação específica, instalada em algum lugar da internet, ou deve prover serviços para qualquer aplicação que solicitar estes serviços? A solução mais comum é um servidor oferecer serviços para qualquer cliente, não um cliente particular (veja a Figura 24.2).
- Quando uma aplicação deve estar rodando, o tempo todo ou apenas quando são necessários os serviços dela? Geralmente, um programa cliente, o qual solicita o serviço, deve rodar somente quando necessário. Por outro lado, o programa servidor, o qual oferece um serviço, deve rodar durante todo o tempo porque o servidor não sabe quando os seus serviços serão necessários.
- Pode haver uma única aplicação universal capaz de oferecer quaisquer serviços que um usuário solicitar ou há apenas uma aplicação para cada tipo de serviço? Na Internet, os serviços são necessários freqüentemente por muitos usuários ao mesmo tempo que dispõem de um programa cliente-servidor específico. Por exemplo, todos nós conhecemos programas criados especificamente para permitir que usuários acessem arquivos ou leiam e-mail. Para serviços ainda mais customizados teremos uma aplicação genérica para permitir aos usuários acessarem os serviços disponíveis em um computador remoto.

Cliente Um **cliente** é um programa, rodando na máquina local, que solicita os serviços de um servidor. Um programa cliente é inicializado por um usuário ou outra aplicação e é finalizado quando o serviço é terminado. Um cliente estabelece um canal de comunicação usando o endereço IP do *host* remoto e o número de porta conhecida do programa servidor específico que deseja acessar. Isto é denominado um **active open**. Após a abertura do canal de comunicação, o cliente envia a solicitação e recebe uma resposta. Embora a solicitação-resposta pode ser repetida muitas vezes, todo o processo é finito e eventualmente chega ao fim. Nesse instante, o cliente fecha o canal de comunicação usando um **active close**.



Figura 24.1 Arquitetura cliente-servidor.

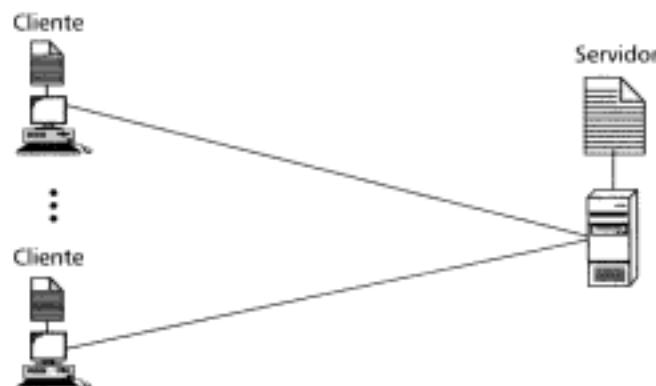


Figura 24.2 Relacionamento cliente-servidor.

Servidor Um **servidor** é um programa, rodando na máquina remota, que oferece serviços aos clientes. Quando inicializado, o programa servidor abre portas de entrada para receber solicitações de clientes, mas nunca inicia um serviço até que seja solicitado a fazê-lo. Isto é denominado um *passive open*.

Um programa servidor deve rodar, teoricamente, sem interrupções. Quando inicializado, o programa servidor começa a rodar esperando pelas solicitações dos clientes. Quando uma solicitação chega, o servidor responde à solicitação, iterativa ou concorrentemente, como veremos adiante.

Concorrência

Tanto o cliente quanto o servidor podem rodar em modo concorrente.

Aplicações Clientes Concorrentes

Os clientes podem rodar numa máquina no modo interativo ou concorrente. Clientes rodando **iterativamente** significa que estão sendo rodados um a um; um cliente inicia, roda e termina antes da máquina iniciar outro cliente. Hoje em dia, a maioria dos sistemas operacionais permite **clientes concorrentes**, isto é, dois ou mais clientes podendo rodar ao mesmo tempo.

Aplicações Servidoras Concorrentes

Um **servidor iterativo** pode processar somente um solicitação por vez. Ele recebe uma solicitação, processa e envia a resposta ao solicitante (um cliente) antes de atender outra solicitação. Por outro lado, um **servidor concorrente** pode processar muitas solicitações ao mesmo tempo, e assim, compartilhar o tempo entre as muitas solicitações.

Os servidores usam o UDP (sem conexão) ou TCP (orientado à conexão) como protocolo da camada de transporte. Assim, a operação do servidor depende de dois fatores: protocolo da camada de transporte e o método de serviço. Teoricamente é possível encontrar quatro tipos de aplicações servidoras: iterativo sem conexão, concorrente sem conexão, iterativo orientado à conexão e concorrente orientado à conexão.

Servidor Iterativo Sem Conexão As aplicações servidoras que usam o UDP normalmente são iterativas, o que, como vimos, significa que o servidor processa uma solicitação por vez. Os servidores usam um único número de porta para este propósito, as portas conhecidas. Todos os pacotes que chegam a esta porta aguardam na fila para serem servidos, conforme ilustra a Figura 24.3.

Servidor Concorrente Orientado à Conexão As aplicações servidoras que usam o TCP normalmente são concorrentes. Isto significa que o servidor pode servir a muitos usuários ao mesmo tempo. A comunicação é orientada à conexão, o que significa que uma solicitação chega em um fluxo de *bytes*, organizados em diversos segmentos, e a resposta pode ocupar muitos segmentos também. Uma conexão é estabelecida entre o servidor e cada cliente e a conexão permanece aberta até que todo o fluxo seja processado.



Figura 24.3 Servidor iterativo sem conexão.

Hidden page

Hidden page

Raw Socket Alguns protocolos como o ICMP ou OSPF que usam diretamente os serviços do IP não utilizam os *stream sockets* nem os *packet sockets*. Os **raw sockets** foram desenvolvidos para estes tipos de aplicação.

Servidor Iterativo sem Conexão

Nesta seção discutiremos a comunicação sem conexão iterativa cliente-servidor usando o UDP e *packet sockets*. Como vimos antes, um servidor que usa o protocolo UDP é usualmente um servidor iterativo sem conexão. Isto significa que o servidor só consegue atender a uma solicitação por vez. Um servidor pega a solicitação recebida em um pacote do UDP, processa a solicitação e gera a resposta para o UDP enviar ao cliente. O servidor não se preocupa com outros pacotes. Estes pacotes, oriundos de um único cliente ou de vários, são armazenado numa fila, aguardando serviço. Eles são processados um a um na ordem de chegada.

O servidor usa apenas um tipo de porta para este propósito: uma porta conhecida. Todos os pacotes chegando nesta porta esperam na fila até serem servidos. A Figura 24.7 mostra o fluxo de eventos em uma comunicação iterativa sem conexão.

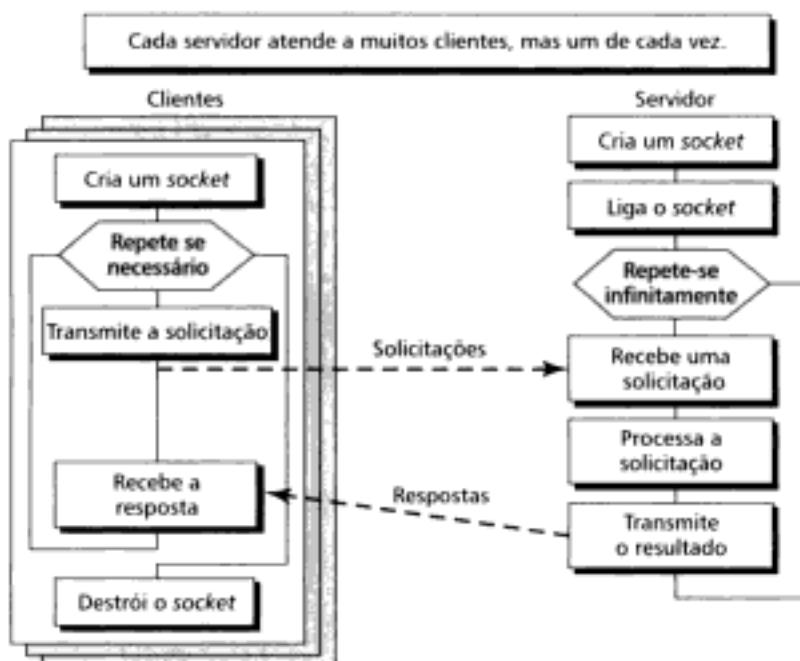


Figura 24.7 Interface socket para o servidor iterativo sem conexão.

Servidor

O servidor realiza as seguintes funções:

1. **Criação de um socket.** O servidor solicita ao sistema operacional a criação de um *socket*.
2. **Bind (ligação).** O servidor solicita ao sistema operacional que entre com a informação no *socket* relacionado ao servidor. Isto recebe o nome de *ligação do socket do servidor*.
3. **Repetição.** O servidor repete infinitamente os seguintes passos:
 - a. **Recebe uma solicitação.** O servidor informa ao sistema operacional que espere por uma solicitação destinada a este *socket* e a receba, quando chegar.
 - b. **Processa.** A solicitação é processada pelo servidor.
 - c. **Transmite.** A resposta é enviada ao cliente.

Cliente

O cliente realiza as seguintes funções:

1. **Criação de socket.** O cliente solicita ao sistema operacional que crie um *socket*. Nesse caso, não há necessidade de fazer ligação. O sistema operacional normalmente adiciona a informação no *socket*.

2. **Repetição.** O cliente repete os seguintes passos à medida que são requeridos:
 - a. **Transmite.** O cliente solicita ao sistema operacional enviar uma solicitação.
 - b. **Recebe.** O cliente solicita ao sistema operacional que espere pela resposta e entregue-a quando chegar.
 - c. **Destrução de socket.** Quando o cliente não tem mais solicitações, ele informa ao sistema operacional que destrua o *socket*.

Servidor Orientado à Conexão

Nesta seção discutiremos a comunicação orientada à conexão concorrente cliente-servidor usando o TCP e o *stream sockets*. Como vimos antes, um servidor que usa o protocolo TCP é usualmente um servidor concorrente orientado à conexão. Isto significa que o servidor consegue atender a vários clientes de uma só vez. A comunicação é orientada à conexão. Assim, uma solicitação chega na forma de um fluxo de *bytes*, provavelmente em vários segmentos, e a resposta também ocuparia vários segmentos. Uma conexão é estabelecida entre o servidor e cada cliente. A conexão permanece ativa até que todo o fluxo seja processado. Em seguida, a conexão é terminada.

O servidor deve possuir um *buffer* para cada conexão. Os segmentos dos clientes são armazenados em *buffers* apropriados e controlados concorrentemente pelo servidor.

Para oferecer este serviço, a maioria das implementações usam o conceito de servidor pai e servidor filho. Um servidor rodando infinitamente e aceitando conexões dos clientes é denominado *servidor pai*. O servidor pai sempre usa uma porta conhecida. Após o estabelecimento da conexão, o servidor pai cria um *servidor filho* e uma porta efêmera é atribuída ao servidor filho para controlar o cliente. A porta conhecida fica livre e pode esperar por outra conexão. Nessa seção, mostraremos como um servidor é capaz de atender a muitos clientes concorrentes, usando os serviços do TCP. A Figura 24.8 mostra o fluxo de eventos para um servidor e um cliente.

Servidor

O servidor realiza as seguintes funções:

1. **Criação de socket.** O servidor solicita ao sistema operacional que crie um *socket*.
2. **Bind (ligação).** O servidor solicita ao sistema operacional que entre com a informação no *socket* criado na etapa anterior.
3. **Listen (escuta).** O servidor solicita ao sistema operacional para ficar passivo e escutar o cliente que necessita de uma conexão com este servidor. Lembre-se que o TCP é um protocolo orientado à conexão. Uma conexão precisa ser estabelecida antes dos dados serem transferidos.
4. **Repetir.** O servidor repete infinitamente os seguintes passos:
 - a. **Criação de servidor filho.** Quando um filho solicita uma conexão, o sistema operacional cria um processo filho temporário e atribui a função de atender o cliente ao filho. O processo pai fica livre para escutar novos clientes.
 - b. **Criação de novo socket.** Um novo *socket* é criado para ser utilizado pelo processo cliente.
 - c. **Repetição.** O servidor filho repete as seguintes etapas à medida que são requeridas:
 - Lê.** O filho lê um fluxo de *bytes* da conexão. Lembre-se que o TCP é um protocolo orientado a *bytes*.
 - Processa.** O filho processa o fluxo de *bytes*.
 - Escreve.** O filho escreve os resultados como um fluxo de *bytes* para a conexão.
 - d. **Destrução do socket.** Tão logo o cliente tenha sido servido, o processo filho solicita ao sistema operacional que destrua o *socket* temporário.

Cliente

O cliente realiza as seguintes funções:

1. **Criação de socket.** O cliente solicita ao sistema operacional que crie um *socket*.
2. **Conexão.** O cliente repete as seguintes etapas à medida que elas são requeridas:
 3. **Escreve.** O cliente envia um fluxo de *bytes* ao servidor.
 - a. **Lê.** O cliente recebe um fluxo de *bytes* do servidor.
 - b. **Destrução do socket.** Tão logo o cliente tenha terminado, solicita ao sistema operacional que destrua o *socket*. A conexão é assim fechada.
 4. **Programas Cliente e Servidor.**

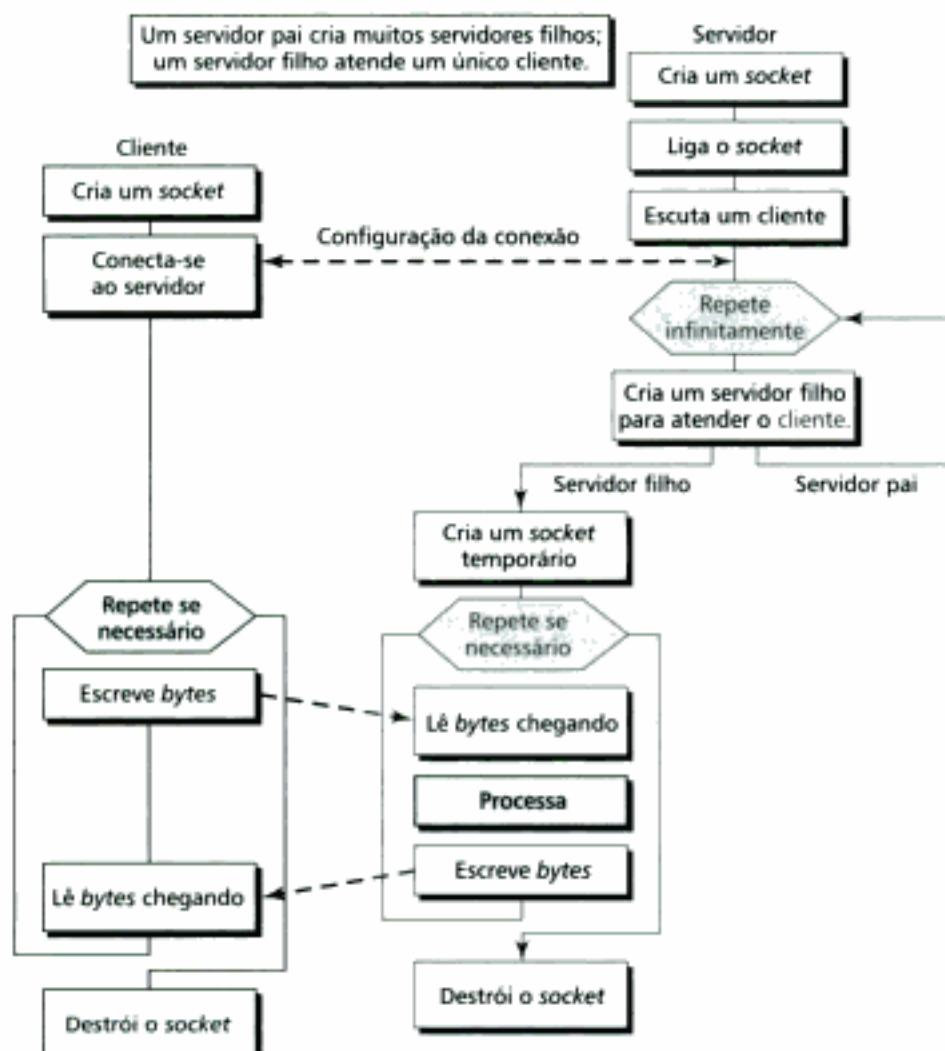


Figura 24.8 Interface socket para o servidor concorrente orientado à conexão.

Programas cliente-servidor são escritos em linguagens tais como C, C++, Java e Perl. Este tipo de programação é muito complicada e requer um conhecimento avançado tanto de programação quanto de linguagens de programação particulares. Estes tópicos infelizmente estão fora do foco deste livro. Entretanto, colocamos alguns exemplos de programas cliente-servidor no Apêndice H para os leitores interessados.

24.3 TERMOS-CHAVE

<i>Active close</i>	Programa aplicativo
<i>Active open</i>	<i>Raw socket</i>
Application Programming Interface (API)	Servidor
Arquitetura cliente-servidor	Servidor concorrente
Cliente	Servidor concorrente orientado à conexão
Cliente concorrente	Servidor iterativo
Cliente iterativo	Servidor iterativo sem conexão
Interface <i>socket</i>	<i>Socket</i>
<i>Packet socket</i>	<i>Stream socket</i>
Processo	

24.4 RESUMO

- Na arquitetura cliente-servidor, o cliente roda um programa que solicita um serviço e o servidor roda um programa que fornece o serviço. Estes dois programas comunicam-se.
- Um programa servidor pode oferecer serviços a muitos programas cliente.
- Clientes podem ser rodados iterativa (um de cada vez) ou concorrentemente (muitos ao mesmo tempo).
- Servidores podem controlar clientes iterativa (um de cada vez) ou concorrentemente (muitos ao mesmo tempo).
- Um servidor iterativo sem conexão usa o UDP como protocolo da camada de transporte e pode servir um cliente de cada vez.
- Um servidor concorrente orientado à conexão usa o TCP como protocolo da camada de transporte e pode servir muitos clientes ao mesmo tempo.
- Quando o sistema operacional executa um programa é criada uma instância do programa, denominada processo.
- Se duas aplicações, uma rodando num sistema local e outra rodando num sistema remoto, precisam se comunicar é necessário um programa de rede.
- A interface *socket* é um conjunto de declarações, definições e procedimentos para escrita de programas cliente-servidor.
- A estrutura de comunicação necessária à programação *socket* é denominada um *socket*.
- Um *stream socket* é usado juntamente com um protocolo orientado à conexão, tal como o TCP.
- Um *packet socket* é usado juntamente com um protocolo sem conexão, tal como o UDP.
- Um *raw socket* é usado por protocolos tal como o ICMP ou OSPF que usam diretamente os serviços do protocolo IP.

24.5 PRATIQUE OS CONHECIMENTOS ADQUIRIDOS

Questões de Revisão

1. Na arquitetura cliente-servidor, discuta os termos cliente e servidor e o relacionamento entre eles.
2. Qual é a diferença entre um *active open* e um *passive open*?
3. Qual é a diferença, em termos de controle, de um servidor que controla clientes iterativamente para um servidor que controla clientes concorrentemente.
4. Como um programa difere de um processo?
5. O que é a interface *socket*?
6. Quais são os componentes de um endereço *socket*?
7. Cite três tipos de interfaces *socket*.
8. Que interface *socket* foi projetada para ser utilizada com o TCP?
9. Como um servidor concorrente orientado à conexão consegue atender a múltiplos clientes ao mesmo tempo?
10. O que é um servidor filho?

Questões de Múltipla Escolha

11. _____ pode solicitar um serviço.
 - Uma interface *socket*
 - Uma porta
 - Um cliente
 - Um servidor
12. _____ pode oferecer um serviço.
 - Um servidor iterativo
 - Um servidor concorrente
13. O programa cliente é _____ porque ele termina após ter sido servido.
 - Ativo
 - Passivo
 - Finito
 - Infinito

14. O programa servidor é _____ porque ele sempre está disponível, esperando por uma solicitação de um cliente.
- Ativo
 - Passivo
 - Finito
 - Infinito
15. Um servidor concorrente orientado à conexão usa portas _____.
- Efêmeras
 - Conhecidas
 - Ativas
 - (a) e (b)
16. Um servidor iterativo sem conexão usa portas _____.
- Efêmeras
 - Conhecidas
 - Ativas
 - (a) e (b)
17. Uma máquina A solicita um serviço X de um máquina B. A máquina B solicita um serviço Y da máquina A. Qual é o número total de programas aplicativos requeridos?
- 1
 - 2
 - 3
 - 4
18. Um cliente envia um _____ quando precisa dos serviços de um servidor.
- Active open*
 - Passive open*
 - Active request*
 - Finite open*
19. Um programa servidor, uma vez que ele envia um _____, espera que os clientes solicitem os serviços dele.
- Active open*
 - Passive open*
 - Active request*
 - Finite open*
20. _____ processa uma solicitação de cada vez.
- Um cliente iterativo
 - Um servidor iterativo
 - Um cliente concorrente
 - Um servidor concorrente
21. _____ processa muitas solicitações ao mesmo tempo.
- Um cliente iterativo
22. Num servidor concorrente orientado à conexão, a _____ é utilizada somente para conexão.
- Porta infinita
 - Porta efêmera
 - Porta conhecida
 - (b) e (c)
23. Um(a) _____ é uma instância de um _____.
- Processo; programa
 - Programa; processo
 - Processo; serviço
 - Estrutura; processo
24. O _____ socket é usado juntamente com um protocolo orientado à conexão.
- Stream*
 - Packet*
 - Raw*
 - Remote*
25. O _____ socket é usado juntamente com um protocolo sem conexão.
- Stream*
 - Packet*
 - Raw*
 - Remote*
26. O _____ socket é usado juntamente com um protocolo que, diretamente, usa os serviços do protocolo IP.
- Stream*
 - Packet*
 - Raw*
 - Remote*
27. Um servidor _____ atende a múltiplos clientes, controlando uma solicitação de cada vez.
- Iterativo orientado à conexão
 - Concorrente orientado à conexão
 - Iterativo sem conexão
 - Concorrente sem conexão
28. Um servidor _____ atende a múltiplos clientes simultaneamente.
- Iterativo orientado à conexão
 - Concorrente orientado à conexão
 - Iterativo sem conexão
 - Concorrente sem conexão

Hidden page

Domain Name System (DNS)

AInternet usa o endereço IP para identificar uma entidade da rede, o qual identifica unicamente a conexão de um *host* na Internet. Entretanto, as pessoas preferem usar nomes em vez de endereços números. Sendo assim, precisamos de um sistema que possa mapear um nome em um endereço e vice-versa.

Quando a Internet era pequena, o mapeamento era feito através de um *host file*. O *host file* tinha somente duas colunas; uma para os nomes e outra para os endereços. Todo *host* podia armazenar o *host file* no próprio disco rígido e atualizá-lo periodicamente a partir de um *host file* mestre. Quando um programa ou usuário necessitava de um mapeamento de um nome em um endereço, o *host* consultava o *host file* e determinava o mapeamento.

Entretanto, hoje é impossível termos um único *host file* relacionando cada endereço a um nome e vice-versa. O *host file* seria grande demais para armazenar em cada *host*. Além disso, seria impossível atualizar todos os *host files* no mundo toda vez que alguma modificação na rede fosse detectada.

Uma solução seria armazenar todo o *host file* em um único computador e permitir o acesso a essa informação centralizada para cada computador que precisasse de mapeamento. Mas, sabemos que isso geraria uma quantidade de tráfego insuportável na Internet.

Outra solução, usada atualmente, é dividir esta quantidade enorme de informações em partes menores e armazená-las em computadores diferentes distribuídos mundo afora. Neste método, o *host* que precisar resolver (vamos começar a usar o *resolver* no lugar do mapear) um nome pode contatar o computador mais próximo que mantém a informação necessária. Este é o método usado pelo **Domain Name System (DNS)**. Neste capítulo, discutiremos inicialmente os conceitos e idéias por trás do DNS. Em seguida, descreveremos o protocolo DNS propriamente dito.

25.1 ESPAÇO DE NOMES

Para evitar ambigüidade, os nomes atribuídos às máquinas (*hosts*) devem ser cuidadosamente selecionados em um **espaço de nomes** com um controle completo sobre a ligação entre os nomes e os endereços IP. Noutras palavras, os nomes devem ser únicos porque os endereços também o são. Um espaço de nomes que mapeia cada endereço em um único nome pode ser organizado de dois modos: plano ou hierárquico.

Espaço de Nomes Plano

Em um **espaço de nomes plano** um nome é atribuído a um endereço. Um nome nesse espaço é uma seqüência de caracteres sem nenhuma estrutura. Os nomes podem ou não ter uma seção comum, se tiver, ele não tem significado. A principal desvantagem do espaço de nomes plano é que ele não pode ser utilizado para resolver nomes em um sistema grande, como a Internet, porque deve ficar centralizado para evitar ambigüidade e duplicação de nomes.

Espaço de Nomes Hierárquico

Em um **espaço de nomes hierárquico**, todo nome é dividido em várias porções. A primeira delas pode definir a natureza da organização, a segunda pode definir o nome, a terceira pode definir o departamento e assim por diante. Nesse caso, a autoridade para atribuir e controlar o espaço de nomes pode ser descentralizada. Uma autoridade central pode atribuir a porção do nome que define a natureza da organização e o nome. A responsabilidade do resto do nome pode ser delegada a organizações e empresas. Sufixos podem ser adicionados ao nome para definir os *hosts* ou recursos. O administrador do sistema não precisa se preocupar com o prefixo escolhido para um determinado *host* porque, até mesmo se outra empresa o tiver adotado, ainda assim o endereço completo será diferente. Por exemplo, suponha que duas pessoas e uma empresa denominem um dos computadores de *challenger*. À primeira pessoa é delegado um nome pela autoridade central (que gerencia o espaço de nomes) de *fhda.edu*; à segunda pessoa é delegado o nome *berkeley.edu* e à empresa é delegado o nome *smart.com*. Quando todos eles organizarem os nomes completos, isto é, incluindo o nome dos computadores (no caso *challenger*), o resultado final será completamente diferente: *challenger.fhda.edu*, *challenger.berkeley.edu* e *challenger.smart.com*. Esses nomes são únicos e não há necessidade de ser atribuído por uma zona de autoridade. A zona de autoridade é responsável por delegar responsabilidade administrativa sobre porções do espaço de nomes a organizações e empresas conectadas na rede.

25.2 ESPAÇO DE NOMES DE DOMÍNIO

O **espaço de nomes de domínio (domain name space)** foi desenvolvido para oferecer uma estrutura hierárquica de nomes de espaço. Neste tipo de hierarquia, os nomes são definidos numa estrutura de árvore invertida, isto é, com a raiz no topo. A árvore pode ter somente 128 níveis: nível 0 (raiz) até o nível 127. Considerando que a raiz no topo une toda a árvore, cada nível da árvore define um nível hierárquico (veja a Figura 25.1).

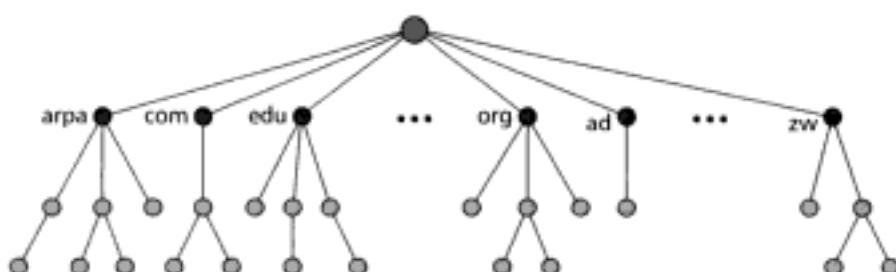


Figura 25.1 Espaço de nomes do domínio.

Componentes do Nome de Nível Superior

Cada nível da árvore possui uma componente do nome, o qual é uma *string* com um máximo de 63 caracteres. O domínio raiz da árvore tem associada uma *string* nula. O DNS requer que as subdivisões do espaço raiz receba componentes do nome diferentes para assegurar a unicidade dos nomes de domínio.

Nome de Domínio

Cada nível (nó) na árvore possui um nome de domínio (um componente do nome). O **nome de domínio** completo é uma sequência de componentes do nome separados por pontos (.). Os nomes do domínio são sempre lidos do nível de interesse mais baixo na direção da raiz. Isto significa que um nome de domínio completo sempre termina numa *string* nula, o que significa que o último caractere é um ponto. A Figura 25.2 mostra alguns nomes de domínios.

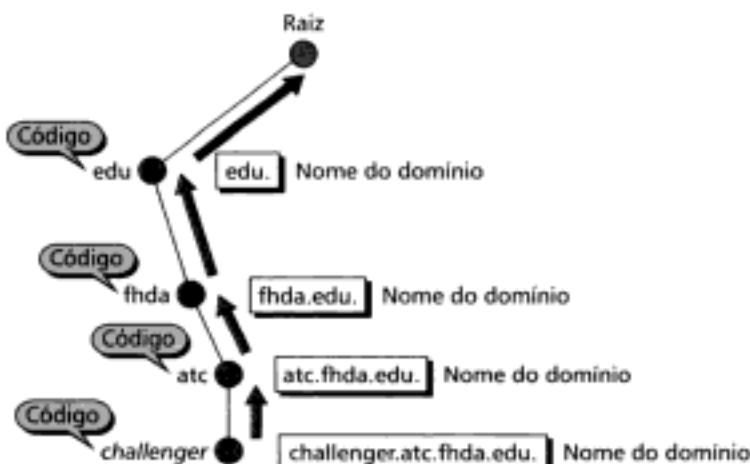


Figura 25.2 Nomes de domínios e códigos.

FQDN

Se um componente do nome termina em uma *string* nula, esse nome é denominado **FQDN (Fully Qualified Domain Name)**. Um FQDN é um nome de domínio que contém o nome completo de um *host*. Ele contém todos os componentes do nome, do mais específico ao mais geral, que definem unicamente o nome do *host*. Por exemplo, o nome do domínio:

challenger.atc.fhda.edu.

é a FQDN de um computador denominado *challenger*, instalado no Advanced Technology Center (ATC) na De Anza College (fhda). Um servidor DNS só pode conter um FQDN para um endereço IP. Observe que o nome deve terminar em um componente nulo e, por isso, terminou em um ponto (.).

PQDN

Se um componente do nome não termina em uma *string* nula, esse nome é denominado **PQDN (Partial Qualified Domain Name)**. Um PQDN é um nome de domínio que é iniciado em um certo componente de nome da hierarquia, mas não alcança a raiz. Ele é utilizado quando o nome a ser resolvido pertence ao mesmo *site* que o cliente. Aqui, resolver significa retirar a porção do nome, denominado **sufixo**, para criar um FQDN. Por exemplo, se um usuário no domínio *fhda.edu* deseja obter o endereço IP de um computador denominado *challenger*, ele ou ela pode definir o nome parcial.

O cliente DNS adiciona o sufixo *atc.fhda.edu* antes de passar o endereço ao servidor de DNS.

O cliente DNS normalmente mantém uma lista de sufixos. A seguir vemos uma lista de alguns sufixos do De Anza College.



A Figura 25.3 mostra alguns FQDNs e PQDNs.

Figura 25.3 FQDN e PQDN.

Domínio

Um **domínio** é uma subárvore do espaço de nomes do domínio. O nome do domínio é a coleção dos componentes de nomes até o topo da subárvore. A Figura 25.4 ilustra alguns domínios. Observe que o próprio domínio pode ser dividido em domínios menores ou **subdomínios**, como às vezes são chamados.

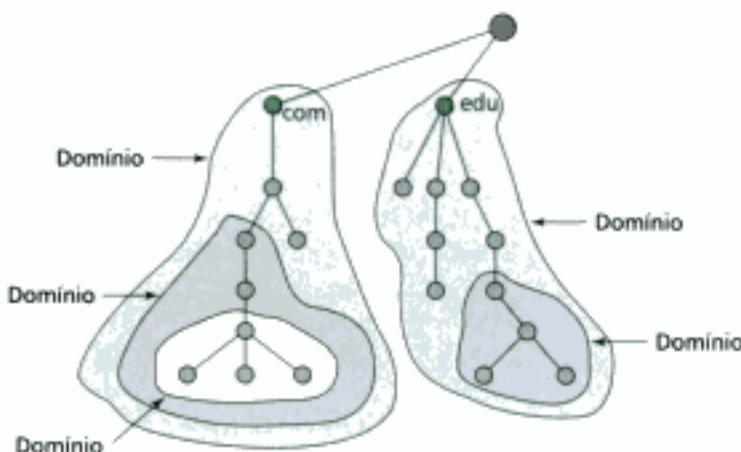


Figura 25.4 Domínios.

25.3 DISTRIBUIÇÃO DO ESPAÇO DE NOMES

A informação armazenada no espaço de nomes do domínio deve ser compartilhada, pois ela é inefficiente e pouco confiável quando fica armazenada em um computador central. Ineficiente porque responder todas as solicitações de todos os lugares gera uma carga pesada de tráfego ao sistema. Pouco confiável porque uma falha no computador central pára completamente o acesso a qualquer tipo de informação na rede.

Hierarquia de Servidores de Nomes

A solução para os dois problemas mencionados é distribuir a quantidade de informação entre muitos computadores denominados **servidores DNS**. Um modo de fazer isso é dividir todo o espaço

de nomes em muitos domínios, todos baseados na raiz. Noutras palavras, deixamos a raiz isolada e criamos muitos domínios (subárvores), todos conectados à raiz. Como o domínio pode crescer e ficar muito grande. O DNS permite a divisão dos domínios em porções menores (os subdomínios). Cada servidor é responsável, i.e., tem autoridade, para quebrar um domínio maior em subdomínios. Assim, é constituída a hierarquia de servidores do mesmo modo que é montada a hierarquia de nomes (veja a Figura 25.5).

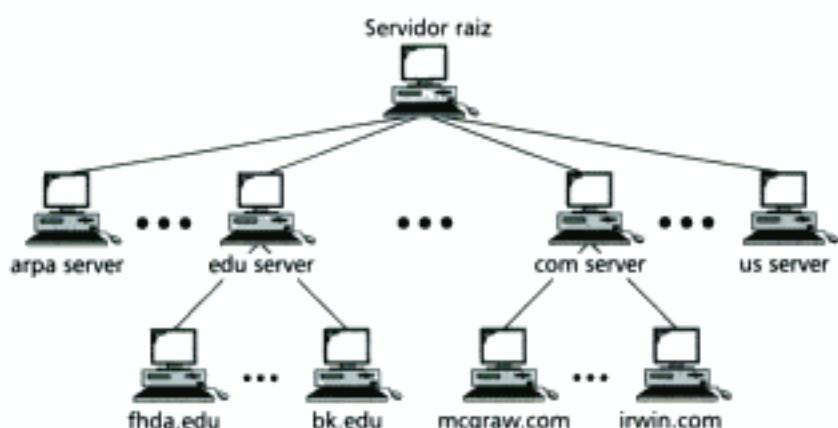


Figura 25.5 Hierarquia de servidores de nomes.

Zonas

Uma **zona** é uma porção administrativa de um domínio DNS. Se um servidor aceita a responsabilidade por um domínio e não o subdivide em domínios menores, o *domínio* e a *zona* são essencialmente a mesma coisa. O servidor constrói um banco de dados, denominado *arquivos de zona*, e mantém nele toda informação relativa ao domínio. Entretanto, se um servidor subdivide um domínio em porções menores (os subdomínios) e delega parte da autoridade a outros servidores, o *domínio* e a *zona* tornam-se coisas diferentes*. A informação sobre os componentes de nomes dos subdomínios é armazenada nos servidores dos níveis mais baixos, com o servidor original mantendo algum tipo referência a esses servidores. É claro que o servidor original não está totalmente livre de responsabilidade: ele ainda administra uma zona, mas o detalhamento da informação é mantida pelos servidores dos níveis inferiores (veja a Figura 25.6).

E o processo continua. Um servidor mais baixo na hierarquia também pode dividir o seu domínio e delegar responsabilidade, mas ainda mantém uma porção do domínio sob a autoridade dele.

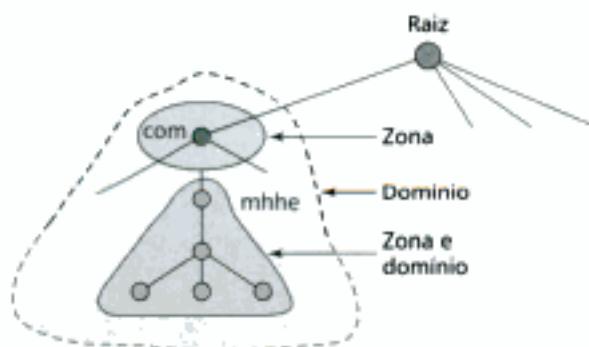


Figura 25.6 Zonas e domínios.

* N. de R. T.: A divisão de um domínio em múltiplas zonas pode ser feita para distribuir tarefas administrativas aos diferentes grupos e fornecer uma distribuição de dados eficiente usando um método de duplicação de dados denominado transferência de zona.

Servidor Raiz

Um **servidor raiz** é um servidor cuja zona consiste de toda a árvore. Um servidor raiz usualmente não armazena qualquer informação sobre os subdomínios, mas delega autoridade a outros servidores, fazendo referência a eles. Atualmente, há 13 servidores raiz, cada qual cobrindo todo o espaço de nomes de domínio. Os servidores raiz estão distribuídos em várias partes do mundo.

Servidores Primários e Secundários

O DNS define dois tipos de servidores: primário e secundário. Um **servidor de nomes primário** é um servidor que armazena arquivos sobre a zona para a qual recebeu autoridade. Ele é responsável por criar, manter e atualizar os arquivos de zona. Ele armazena os arquivos de zona no disco local.

Um **servidor de nomes secundário** é um servidor que transfere toda a informação sobre uma zona de outro servidor (primário ou secundário) e armazena o arquivo em seu disco local. O servidor DNS secundário não cria e nem atualiza os arquivos de zona. Se ocorrer algum tipo de atualização, primeiramente ela deve tomar lugar no servidor primário, o qual transmite a versão atualizada ao secundário. Em suma, o servidor DNS secundário é um servidor de *backup* das informações de zona.

Os servidores primário e secundário têm ambos autoridade sobre as zonas que eles servem. A idéia não é colocar o servidor secundário em um nível de autoridade inferior, mas criar uma redundância de dados de tal forma que se o servidor primário falhar, os outros têm plenas condições de servir os clientes. Observe também que um servidor pode, ao mesmo tempo, ser um servidor DNS primário para uma zona específica e um servidor DNS secundário para outra zona. Sendo assim, quando nos referirmos a um servidor DNS como primário ou secundário, devemos tomar cuidado com a zona que estamos nos referindo.

Um servidor DNS primário contém, no disco rígido, toda a informação da zona sob sua responsabilidade; o servidor DNS secundário armazena, na forma de *backup*, toda a informação da zona do servidor primário.

25.4 DNS NA INTERNET

O DNS é um protocolo utilizado em praticamente todas as plataformas de rede*. Na Internet, a árvore de domínio (espaço de nomes do domínio) foi dividida em três seções diferentes: domínios genéricos ou organizacionais, domínios geográficos e domínios de reserva (veja a Figura 25.7).

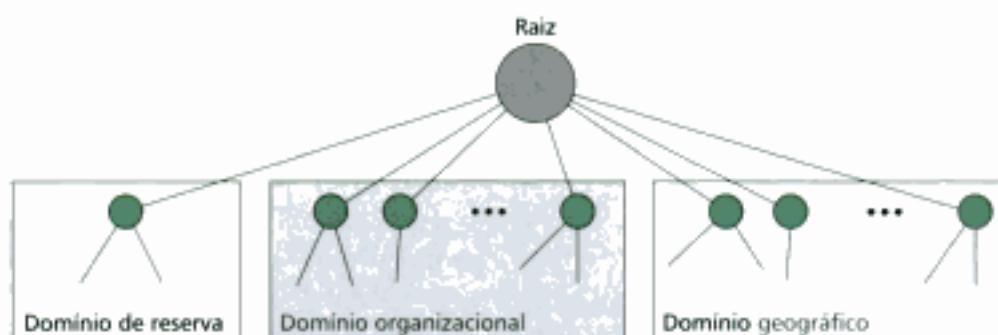


Figura 25.7 DNS na Internet.

* N. de R. T.: Outra forma de DNS foi definido através das RFCs 882, 883 e 973.

Domínios Genéricos

Os **domínios genéricos**, também conhecidos como domínios organizacionais, definem *hosts* registrados de acordo com o comportamento genérico deles. Cada nível hierárquico na árvore define um domínio, o qual usa um código índice de 3 caracteres para indicar a funcionalidade primária contida dentro dele (veja a Figura 25.8).

Olhando para a árvore, vemos que o primeiro nível de uma seção dos domínios genéricos permite sete índice de três caracteres. Estes índices descrevem o tipo de organização, conforme listado na Tabela 25.1.

Recentemente, novos níveis foram adicionados ao primeiro nível. A Tabela 25.2 mostra esses níveis.

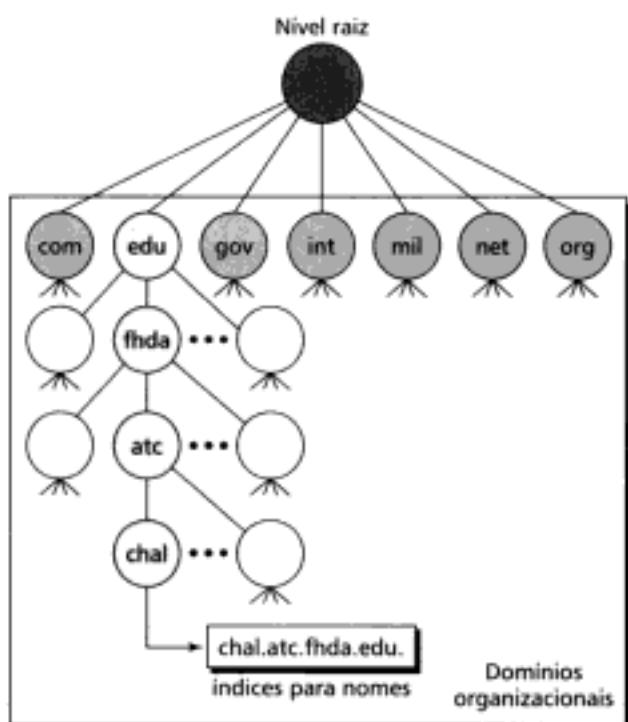


Figura 25.8 Domínios organizacionais.

TABELA 25.1 Códigos para domínios organizacionais

<i>Código</i>	<i>Descrição</i>
com	Empresas ou organizações comerciais
edu	Instituições de ensino
gov	Instituições governamentais
int	Organizações internacionais
mil	Grupos militares
net	Organizações de redes
org	Organizações não comerciais

Domínios Geográficos

A seção de **domínios geográficos** segue o mesmo formato dos domínios genéricos, mas usa somente dois caracteres para representar os códigos dos países (p.ex., *br* para Brasil) no lugar do caractere organizacional. O segundo código pode ser organizacional ou ainda mais específico, estados, província ou regiões de um país. Nos Estados Unidos, por exemplo, é usada a abreviação de estados como uma porção do domínio geográfico *us* (p. ex., *ca.us*).

TABELA 25.2 Novos códigos para domínios organizacionais

Código	Descrição
aero	Empresas áreas/aeroespaciais
biz	Negócios (similar ao com)
coop	Negócios cooperativos
info	Provedores de serviços
museum	Museus e outras organizações não comerciais
name	Nomes pessoais (indivíduos)
pro	Organizações profissionais

A Figura 25.9 mostra uma seção de domínios geográficos. O endereço *anza.cup.ca.us* pode ser traduzido como De Anza College, localizado em Cupertino, Califórnia, Estados Unidos.

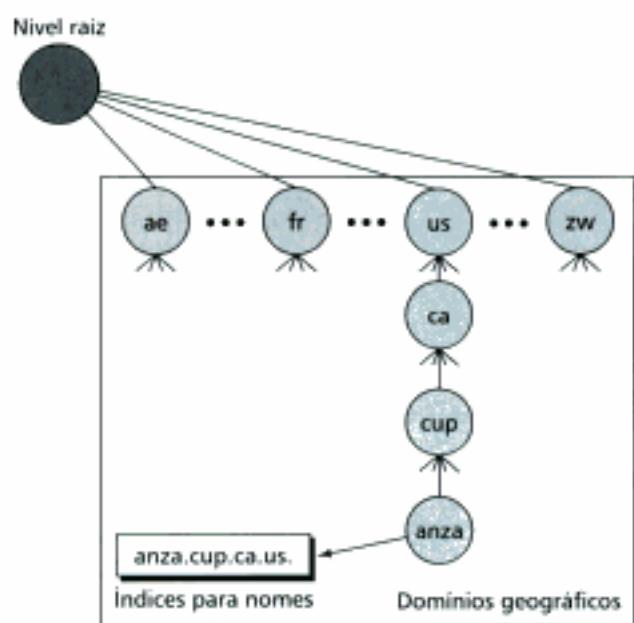


Figura 25.9 Domínios geográficos.

Domínios de Reserva

Os **domínios de reserva** ou **domínios de pesquisa inversa** são usados no mapeamento dos endereços IP em nomes. Isto pode acontecer, por exemplo, quando um servidor recebe de um cliente uma consulta para realização de uma tarefa. Considerando que o servidor mantém uma lista de clientes autorizados, o servidor lista somente o endereço IP do cliente (extraído do pacote IP recebido). Para determinar se o cliente está na lista autorizada, o servidor envia uma consulta de pesquisa inversa ao servidor DNS e pede o mapeamento do endereço IP em nome.

Este tipo de consulta é denominada *ponteiro PTR*. Para controlar o ponteiro, o domínio de reserva é adicionado ao espaço de nomes de domínio no primeiro nível da hierarquia, denominado *arpa* (por razões históricas). O segundo nível da hierarquia também é definido por um único conjunto de caracteres, denominados *in-addr* (endereço inverso). O resto do domínio define os endereços IP.

Os servidores que controlam o domínio de reserva também são hierárquicos. Isto significa que a parte de *netid* do endereço IP deve estar em um nível mais alto que a parte de sub-rede que, por sua vez, deve estar em um nível mais alto que a parte de *hostid*. Desse modo, um servidor que atende ao *site* como um todo ficará em um nível mais elevado que os servidores internos a cada

sub-rede. Esta configuração faz a pesquisa de nomes reserva, se comparada à pesquisa realizada nos domínios genérico ou geográfico. Para seguir a convenção de leitura dos códigos identificadores de domínio, escreva o endereço de baixo para cima. Por exemplo, um endereço IP classe B 132.34.45.121 (parte de $netid = 132.34$) deve ser lido na zona de pesquisa inversa como 121.45.34.132.in-addr.arpa. Observe na Figura 25.10 uma ilustração da configuração do domínio de reserva.

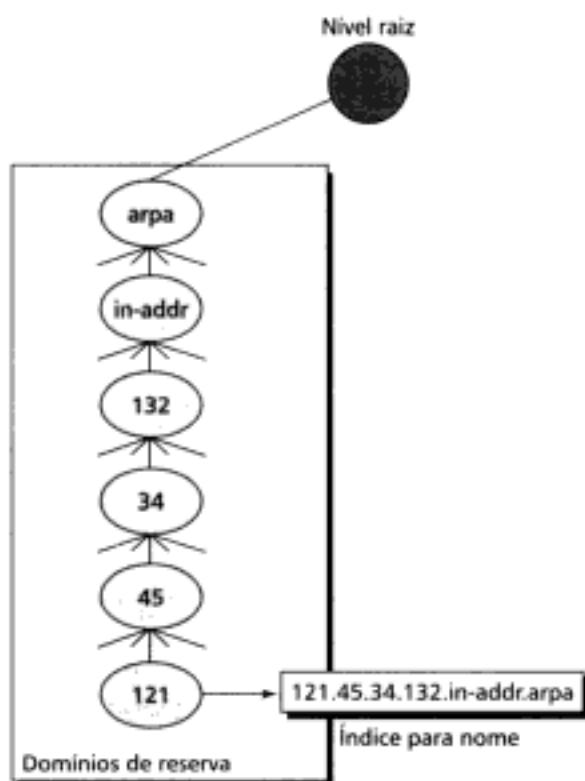


Figura 25.10: Domínio de reserva.

25.5 RESOLVENDO NOMES

O mapeamento de um nome em endereço IP ou vice-versa é denominado **resolução de nomes de endereço**.

Resolver

O DNS é designado como uma aplicação cliente-servidor. Um *host* que precisar dos serviços de resolução de nome em endereço IP ou vice-versa aciona uma aplicação cliente DNS denominada **resolver**. O *resolver* acessa o servidor DNS mais próximo com uma consulta de mapeamento. Se o servidor tiver a informação ele a entrega ao *resolver*. Caso contrário, ele entrega a consulta a outros servidores DNS da rede para que algum deles resolva nome em IP ou vice-versa.

Após o *resolver* receber a resposta do mapeamento, ele a interpreta e verifica se a resolução existe de fato ou se ocorreu um erro, para então, finalmente, entregar o resultado ao processo que fez a consulta.

Mapeando Nomes em Endereços

Na maioria das vezes, o *resolver* fornece um nome de domínio ao servidor e solicita pelo endereço IP correspondente. Nesse caso, o servidor verifica o domínio genérico (organizacional) ou domínio geográfico para determinar o mapeamento.

Se o nome do domínio vem da seção de domínios genéricos, o resolver recebe um nome de domínio, como *chal.atc.fhda.edu*, e solicita ao servidor DNS local que resolva o endereço. Se o servidor local não puder resolver a consulta, ele repassa ao *resolver* de outros servidores DNS ou pergunta diretamente a algum dos servidores DNS que ele tiver conhecimento na rede.

Se o nome do domínio vem da seção de domínios geográficos, o *resolver* recebe um nome de domínio, como *ch.fhda.cu.ca.us*, e repete exatamente o mesmo procedimento citado acima.

Mapeando Endereços IP em Nomes

Um cliente pode enviar um endereço IP para que um servidor DNS faça o mapeamento para nome do domínio. Como mencionado antes, esta consulta acontece através de um ponteiro PTR. Para responder a consultas desse tipo, o DNS usa o domínio de reserva (domínio de pesquisa inversa). Entretanto, numa consulta, o endereço IP é revertido em dois códigos, *in-addr* e *arpa*, que são adicionados para criar um domínio aceitável pela seção do domínio de pesquisa inversa. Por exemplo, se o resolver receber o endereço IP 132.34.45.121, primeiro inverte a ordem dos endereços, e então, adiciona os dois códigos antes de enviá-lo. Nesse caso, o nome de domínio enviado será *121.45.34.132.in-addr.arpa*, o qual será recebido pelo servidor DNS local e resolvido.

Resolução Recursiva

O cliente (*resolver*) pode pedir uma pesquisa recursiva a um servidor de nomes. Isto significa que o resolver espera que o servidor forneça a resposta final. Se o servidor tem autoridade sobre o nome do domínio solicitado, ele verifica a base de dados e responde. Caso contrário, o servidor retransmite a consulta para outro servidor (usualmente um servidor DNS pai) e aguarda a resposta. O servidor DNS pai responde, se ele tiver autoridade sobre o nome do domínio solicitado. De outro modo, ele retransmite a consulta para outro servidor. Quando a consulta tiver sido finalmente resolvida, a resposta é enviada de volta ao cliente que iniciou a consulta. Esta é a **resolução recursiva**, conforme ilustra a Figura 25.11.

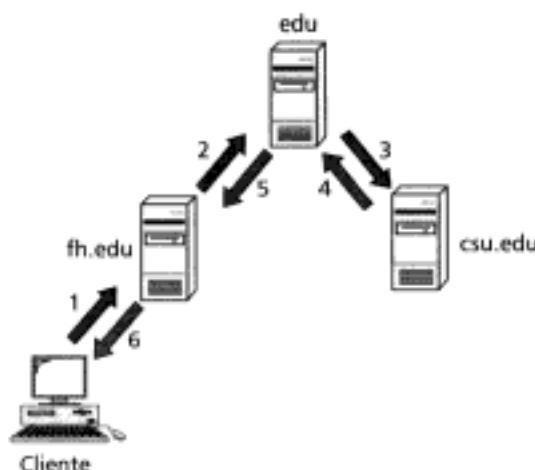


Figura 25.11 Resolução recursiva.

Resolução Iterativa

Se o cliente não pedir uma resolução recursiva, o mapeamento pode ser feito iterativamente. Se o servidor local tiver autoridade sobre o nome solicitado, ele responde. Caso não, o servidor local retorna uma resposta ao cliente contendo o endereço IP de um outro servidor DNS que ele (o servidor local) acha que talvez tenha condições de responder à consulta do cliente. Assim, o cliente repete a consulta, dessa vez perguntando ao servidor DNS sugerido pelo servidor local. Se o novo servidor endereço puder resolver, o endereço é retornado ao endereço IP desejado. Caso contrário, es-

se servidor envia ao cliente o endereço IP de um outro servidor DNS (o terceiro) para uma nova pesquisa. Nesse caso, o cliente repete a consulta ao terceiro servidor. Este processo é denominado **resolução iterativa** porque o cliente repete a mesma consulta a vários servidores DNS. Na Figura 25.12, um cliente solicita uma resolução de nomes a três servidores DNS antes de obter uma resposta IP do servidor *csu.edu*.

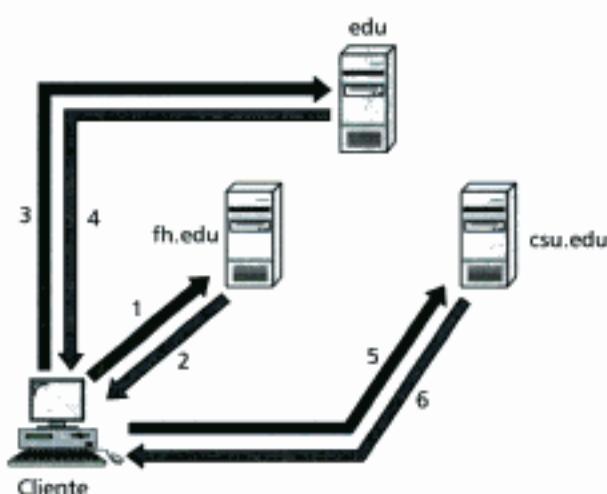


Figura 25.12: Resolução iterativa.

Caching

Embora todos os servidores DNS façam o *cache* de consultas que eles tenham resolvido, os servidores que usam **caching** são servidores DNS que também executam consultas, armazenam as respostas em *cache* e retornam os resultados. Em outras palavras, eles não são autorizados para todos os domínios e apenas armazenam dados que ficam em *cache* enquanto resolvem as consultas. Quando um servidor solicita um mapeamento de outro servidor DNS e recebe a resposta, ele armazena essa informação na memória *cache* antes de enviá-la ao cliente. Se o mesmo ou outro cliente solicitar pelo mesmo mapeamento, esse servidor verifica no *cache* e resolve o problema. Entretanto, para informar ao cliente que a resposta está chegando de um *cache* e não de uma fonte autorizada, o servidor marca a resposta como *unauthoritative*.

O mecanismo de *caching* aumenta a velocidade da resolução, mas também pode se tornar problemática. Se um servidor armazena o mapeamento por muito tempo no *cache*, pode ser que o mapeamento tenha ficado desatualizado. São utilizadas duas técnicas para solucionar este problema. Primeira, o servidor autorizado sempre adiciona uma porção da informação ao mapeamento denominado *Time-To-Live (TTL)*. O TTL define o tempo, em segundos, para o qual o servidor de *caching* pode manter a informação. Após este tempo, o mapeamento é invalidado e qualquer consulta nova deve ser enviada novamente a um servidor autorizado. Segundo, o DNS requer que cada servidor mantenha um contador TTL para cada mapeamento em *cache*. A memória *cache* deve ser pesquisada periodicamente à procura de mapeamentos que tenham expirado o TTL. Nesse caso, estes mapeamentos são apagados da memória.

25.6 MENSAGENS DNS

O DNS possui dois tipos de mensagens: mensagem de consulta (*query*) e a mensagem de resposta (*response*). Ambos tipos de mensagem têm o mesmo formato. A **mensagem de consulta** consiste em um cabeçalho e a seção onde é inserida a pergunta. A **mensagem de resposta** consiste em um cabeçalho, uma seção resposta, uma seção autoridade e uma seção com as informações adicionais (veja a Figura 25.13).

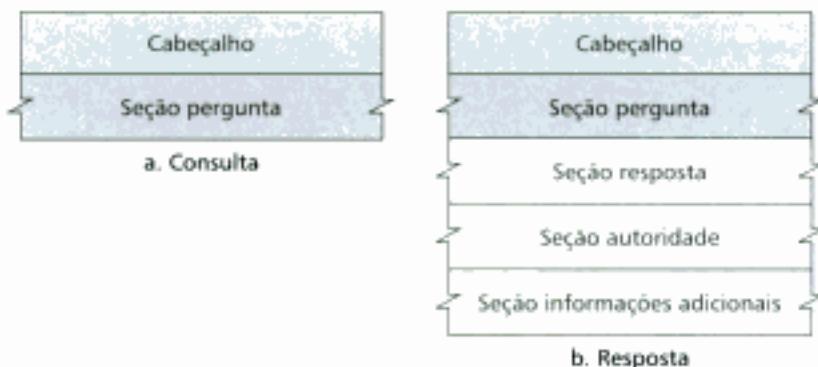


Figura 25.13 Mensagens de consulta e de resposta.

Cabeçalho

Tanto a mensagem de consulta quanto a resposta tem o mesmo formato de cabeçalho, com os *bits* de algumas seções em zero para a mensagem de consulta. O cabeçalho tem um tamanho de 12-bytes e formato conforme ilustra a Figura 25.14.

		2 bytes	2 bytes	
		Identificação	Flags	
	Número de registro de perguntas		Número de registro de respostas (todos os bits em zero na mensagem de consulta)	
	Número de registro de autoridades (todos os bits em zero na mensagem de consulta)		Número de registro de info. adicionais (todos os bits em zero na mensagem de consulta)	

Figura 25.14 Formato do cabeçalho.

A seção de *identificação* é utilizada pelo cliente para identificar a resposta a uma determinada consulta. O cliente usa um número de identificação diferente toda vez que ele envia uma consulta. O servidor, por sua vez, copia este número na resposta correspondente. Os *flags* são uma coleção de subcampos que definem o tipo de mensagem, o tipo de resposta solicitada, o tipo de resolução desejada (recursiva ou iterativa) e assim por diante. A seção de *número de registros de perguntas* contém a quantidade de consultas gravadas na seção pergunta da mensagem. Já a seção *número de registro de respostas* contém a quantidade de respostas gravadas na seção resposta. A seção *número de registro de autoridades* contém a quantidade de servidores autorizados gravados na seção autorizado. Finalmente, a seção *número de registro de informações adicionais* contém a quantidade de registros adicionais no campo adicional de uma mensagem de resposta. Os *bits* das seções *registro de respostas*, *registro de autoridades* e *registro de informações adicionais* são todos zero.

Seção Pergunta

Esta é uma seção que consiste de um ou mais registros de perguntas. Ela é encontrada tanto na mensagem de consulta quanto na mensagem de resposta.

Seção Resposta

Esta é uma seção que consiste de um ou mais registros de recursos. Ela se faz presente somente nas mensagens de resposta. Esta seção inclui a resposta do servidor ao cliente (*resolver*).

Seção Autoridade

Esta é uma seção que consiste de um ou mais registros de recursos. Ela se faz presente somente nas mensagens de respostas. Esta seção fornece informação (nome do domínio) sobre um ou mais servidores autorizados para a consulta.

Seção Informação Adicional

Esta é uma seção que consiste de um ou mais recursos. Ela se faz presente somente nas mensagens de resposta. Esta seção fornece informação adicional ao *resolver*. Por exemplo, um servidor pode fornecer, na seção autoridade, o nome do domínio de um servidor autorizado ao *resolver* e pode incluir o endereço IP do mesmo servidor autorizado nessa seção.

25.7 DDNS

Quando o DNS foi projetado, ninguém previu que a Internet seria tão dinâmica, suportando tantas mudanças de endereços ao mesmo tempo. No DNS, quando acontece uma mudança – tal como adição ou remoção de um novo *host* a rede ou mudança de endereço IP – essa informação deve ser feita no *arquivo mestre de nomes*. Tais mudanças envolvem uma quantidade enorme de atualização manual. A dimensão atual da Internet não permite este tipo de operação.

A solução é: o arquivo mestre deve ser atualizado dinamicamente. O **Dynamic Domain Name System (DDNS)** foi criado para responder a essa necessidade. No DDNS, quando a ligação entre um nome e um endereço IP é determinada, a informação é enviada, usualmente pelo DHCP (veja Capítulo 19), ao servidor DNS primário. Assim, o servidor primário atualiza a zona. Os servidores secundários são notificados ativa ou passivamente. Na notificação ativa, o servidor primário envia uma mensagem aos servidores secundários sobre a mudança na zona. Considerando a notificação passiva, os servidores secundários devem verificar periodicamente quaisquer mudanças na rede. Nesse caso, após serem notificados sobre a mudança, o servidor secundário solicita informação sobre toda a zona, isto é, pede a transferência de zona.

25.8 ENCAPSULAMENTO

O DNS pode usar tanto o UDP quanto o TCP. Em ambos casos, a porta conhecida de DNS, número 53, é usada pelo servidor. O UDP é utilizado quando o tamanho da mensagem de resposta é menor que 512 *bytes* porque a maioria dos pacotes UDP tem um limite de 512 *bytes*. Se o tamanho da mensagem resposta for maior que 512 *bytes*, uma conexão TCP deve ser utilizada. Nesse caso, pode acontecer um dos seguintes cenários:

1. Se o *resolver* faz idéia que o tamanho da mensagem de resposta é superior a 512 *bytes*, ele realiza uma conexão TCP. Por exemplo, se um servidor de nomes secundário (agindo como cliente) necessitar de uma transferência de zona, ele deve usar uma conexão TCP porque o tamanho da informação a ser transferida certamente excederá os 512 *bytes*.
2. Se o *resolver* não faz idéia do tamanho da mensagem de resposta, ele pode tentar usar a porta UDP. Entretanto, se o tamanho da mensagem de resposta superar os 512 *bytes*, o servidor trunca a mensagem. Então, o *resolver* reabre a conexão, nesse caso usando o protocolo TCP, e repete a solicitação na tentativa de obter uma resposta completa do servidor.

O DNS pode usar tanto os serviços do UDP quanto os serviços do TCP através da porta número 53.

Hidden page

25.11 PRATIQUE OS CONHECIMENTOS ADQUIRIDOS

Questões de Revisão

1. Qual é a vantagem da hierarquia de espaços de nomes sobre o espaço de nomes planos para um sistema do tamanho da Internet?
2. Qual é a diferença entre um servidor de nomes primário e um servidor de nomes secundário?
3. Quais são os três domínios de um espaço de nomes do domínio?
4. Qual é o propósito do domínio de reserva?
5. Como a resolução recursiva difere da resolução iterativa?
6. O que é FQDN?
7. O que é PQDN?
8. O que é uma zona DNS?
9. De que forma o mecanismo de *caching* aumenta a eficiência da resolução de nomes?
10. Quais são os dois tipos principais de mensagens DNS?
11. Por que o DDNS teve de ser desenvolvido?

Questões de Múltipla Escolha

12. No domínio *chal.atc.fhda.edu*, _____ é o último código específico.
 - chal*
 - atc*
 - fhda*
 - edu*
13. No domínio *chal.atc.fhda.edu*, _____ é o código mais específico.
 - chal*
 - atc*
 - fhda*
 - edu*
14. Qual dos seguintes nomes de domínio usa um domínio geográfico para ser resolvido em endereço IP?
 - chal.atac.fhda.edu*
 - gsfc.nasa.gov*
 - kenz.eng.sony.jp*
 - mac.eng.sony.com*
15. Uma resposta DNS é classificada como _____ se a informação chega de uma memória *cache*.
 - Authoritative*
 - Unauthoritative*
 - Iterativa
 - Recursiva
16. Na resolução _____, o cliente está em contato direto com mais de um servidor DNS.
 - Recursiva
 - Iterativa
 - Em *caching*
 - Todas as alternativas anteriores
17. Na resolução _____, o cliente poderia contatar diretamente mais de um servidor DNS.
 - Recursiva
 - Iterativa
 - Em *caching*
 - Todas as alternativas anteriores
18. Na resolução de um endereço IP em um nome é utilizado o domínio _____.
 - De reserva
 - Reverso
 - Genérico ou organizacional
 - Geográfico
19. Como o tempo de vida de uma resolução de nome em endereço IP é controlado na memória cache?
 - Através do campo TTL configurado pelo servidor
 - Através do contador TTL configurado pelo servidor

Hidden page

Correio Eletrônico (SMTP) e Transferência de Arquivo (FTP)

Há duas aplicações bastante populares para a troca de informações. O correio eletrônico (*e-mail*) é um aplicativo de rede usado para enviar mensagens de correio eletronicamente através de vários tipos de redes usando os mais diversos protocolos. A transferência de arquivo é feita por um aplicativo de rede que permite aos computadores trocarem arquivos de um dispositivo em uma rede para outro.

26.1 CORREIO ELETRÔNICO

Um dos serviços de rede mais populares é o correio eletrônico (*e-mail*). O correio eletrônico é usado na transmissão de uma única mensagem que pode incluir: texto, voz, vídeo, gráfico ou mais de um desses objetos. O protocolo **SMTP (Simple Mail Transfer Protocol)** é o mecanismo padrão de correio eletrônico da Internet.



Figura 26.1 Formato de um e-mail.

Envelope

O envelope contém usualmente os endereços do transmissor e do receptor e outras informações que se fizerem necessárias.

Mensagem

A mensagem possui os *cabeçalhos* e o *corpo*. Os **cabeçalhos** da mensagem definem o transmissor, o receptor, o assunto da mensagem e outras informações. O corpo da mensagem contém a informação de fato, a ser lida pelo receptor do *e-mail*.

Recebendo Correio

O sistema de *e-mail* verifica periodicamente a caixa de correio. Se um usuário tiver um *e-mail*, novo ou não, o sistema o destaca usando algum símbolo de advertência. Quando o usuário estiver pronto para ler o *e-mail*, uma lista de entradas é mostrada, onde cada linha contém um sumário de informações sobre uma mensagem particular na caixa de correio. Usualmente, o sumário inclui, dentre outros itens, o endereço de *e-mail* do transmissor, o assunto e a hora em que o *e-mail* foi enviado ou recebido. O usuário pode selecionar qualquer uma das mensagens e exibir o conteúdo dela na tela do computador.

Endereços

Para entregar um *e-mail*, o sistema de correio eletrônico deve usar um esquema de endereçamento com um endereço único. O esquema de endereçamento usado pelo SMTP consiste em duas partes: uma *parte local* e um *nome de domínio*, separados pelo sinal @ (veja a Figura 26.2).

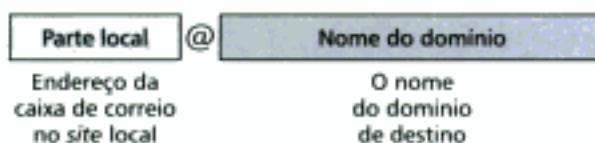


Figura 26.2 Endereço de *e-mail*.

Parte Local

A parte local define o nome de um arquivo especial, denominado caixa de correio, onde todos os *e-mails* recebidos pelo usuário são depositados para serem recuperados pelo *user agent*.

Nome do Domínio

A segunda parte do endereço é o **nome do domínio**. Uma organização seleciona usualmente um ou dois *hosts* da rede para transmitir e receber os *e-mails*. Cada um desses *hosts* é denominado *servidor de e-mail*. O nome do domínio atribuído a cada servidor de *e-mail* é recebido da base de dados DNS ou é um nome lógico (p. ex., o nome da organização).

User Agent (UA)

O primeiro componente de um sistema de correio eletrônico é o **User Agent (UA)**. Às vezes, um *user agent* é chamado *leitor de correio eletrônico*, mas a terminologia é confusa. Nós preferimos o uso do termo *user agent*.

Serviços Oferecidos pelo UA

Um *user agent* é um pacote de *software* (um programa) que compõe, lê, responde e encaminha mensagens. Ele também controla a caixa de correio. A Figura 26.3 mostra os serviços típicos de um *user agent*.



Figura 26.3 User agent.

Compondo Mensagens Um *user agent* é responsável pelo espaço de composição de mensagem de *e-mail*. A maioria dos *user agents* oferecem uma área de texto na tela onde o usuário preenche a bel prazer. Alguns têm até mesmo um editor de textos nativo que permite fazer correção de erros, verificação gramatical e outras tarefas típicas que esperamos de um processador de textos sofisticado. É claro que um usuário pode utilizar um processador de textos de sua preferência para criar a mensagem e, então, importar ou copiá-la no *user agent*.

Lendo Mensagens O segundo serviço de um *user agent* é ler as mensagens de entrada. Quando um usuário chama o *user agent*, primeiro ele verifica os *e-mails* na caixa de entrada. A maioria dos *user agents* mostram numa linha um sumário de cada mensagem recebida, o qual contém os seguintes campos:

1. Número.
2. *Flag*, que mostra se o *e-mail* é novo, se já foi lido e não respondido, lido e respondido e assim por diante.
3. Tamanho da mensagem.
4. Remetente da mensagem.
5. Assunto, se a linha de assunto foi preenchida pelo remetente.

Respondendo Mensagens Após a leitura da mensagem, um usuário pode querer usar o UA para enviar uma resposta. Normalmente, um *user agent* permite que o usuário responda ao remetente original ou a todos usuários que receberam uma cópia da mensagem. A mensagem de resposta normalmente contém a mensagem original (referência rápida) e a nova mensagem.

Encaminhando Mensagens A resposta é definida como uma mensagem enviada ao remetente original ou aos usuários da cópia. O encaminhamento significa enviar a mensagem a uma terceira parte. Um *user agent* permite que o receptor encaminhe a mensagem, com ou sem comentários extra, para outro destinatário.

Controlando a Caixa de Correio Um *user agent* normalmente cria duas caixas: uma de entrada e outra de saída. Cada caixa é um arquivo com um formato especial que pode ser controlado pelo *user agent*. A caixa de entrada mostra todos os *e-mails* recebidos pelo usuário, desde a primeira entrada, até que sejam apagados. A caixa de saída mostra todos os *e-mails* enviados pelo usuário, desde a primeira saída, até que sejam apagados.

Tipos de User Agents

Há dois tipos de *user agents*: orientado a comandos e baseado em interface gráfica GUI.

Orientado a Comandos Os *user agents* orientados a comandos foram desenvolvidos nos primórdios do correio eletrônico. Eles ainda estão presentes como *user agents* subjacentes em servidores. Um *user agent* orientado a comandos normalmente aceita um único caractere do teclado para realizar uma tarefa. Por exemplo, um usuário pode digitar o caractere *r*, na linha de comando do *prompt*, para responder somente ao remetente da mensagem ou digitar o caractere *R* para respon-

der a todos os usuários da lista de recepção. Alguns exemplos de *user agents* orientados a comandos são *mail*, *pine* e *elm*.

Alguns exemplos de *user agents* orientados a comandos são *mail*, *pine* e *elm*.

Baseado em Interface Gráfica GUI Os *user agents* modernos são baseados em interfaces gráficas GUI. Eles possuem componentes gráficos baseados na Graphical User Interface (GUI) que permitem que o usuário interaja com o *software* usando tanto o teclado quanto o *mouse*. Esses componentes gráficos são os ícones, barras de menus e janelas que tornam os serviços fáceis de acessar. Alguns exemplos de *user agents* baseados em interfaces GUI são o Eudora, o Outlook da Microsoft e o Netscape.

Alguns exemplos de *user agents* baseados em interfaces GUI são o Eudora, o Outlook Express e o Netscape.

MIME

O SMTP é um protocolo de transferência de *e-mail* simples. Entretanto, essa simplicidade tem um preço. O SMTP pode enviar mensagens somente no formato de 7-bits do código ASCII. Logo, é inevitável que haja limitações. Por exemplo, ele não pode ser usado em idiomas não suportados pelo código ASCII (tal como o francês, alemão, russo, chinês e o japonês). Além disso, não pode ser utilizado para enviar arquivos binários (arquivos que armazenam dados em uma cadeia de 0s e 1s sem usar qualquer tipo de codificação), áudio ou vídeo.

A especificação **MIME (Multipurpose Internet Mail Extensions)** é um protocolo suplementar que permite transmissões de dados não ASCII através do SMTP. O MIME não é um protocolo de *e-mail* e não pode substituir o SMTP. Ele é apenas uma extensão do SMTP.

O MIME converte dados no formato não ASCII do remetente em dados ASCII e os entrega ao cliente SMTP para ser enviado à Internet. O servidor SMTP recebe esses dados no formato ASCII e os entrega ao MIME para serem reconvertisdos para a forma original.

Podemos pensar no MIME como um conjunto de funções de *software* que converte dados genéricos em dados no formato ASCII e vice-versa (veja a Figura 26.4).

O MIME define cinco tipos de cabeçalho que podem ser adicionados à seção do cabeçalho das mensagens SMTP original para definir os parâmetros de conversão.

1. MIME-Version (MIME-Versão)
2. Content-Type (Conteúdo-Tipo)
3. Content-Transfer-Encoding (Conteúdo-Transferência-Codificação)
4. Content-Id (Conteúdo-Identificação)
5. Content-Description (Conteúdo-Descrição)

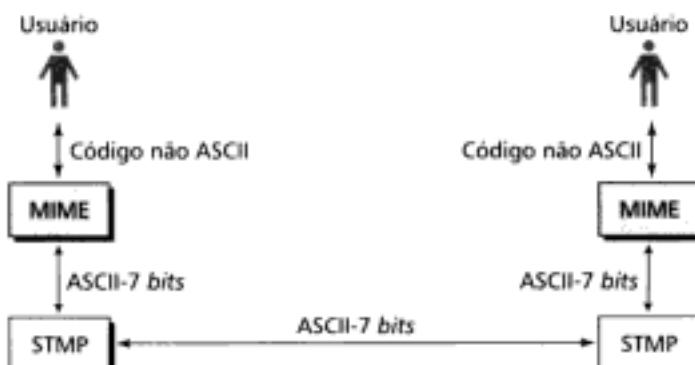


Figura 26.4 User agent.

Hidden page

- **Text.** A mensagem original está no formato ASCII (7-bits) e não é necessário transformá-la. O único *subtype* usado atualmente é o *plain*.
- **Multipart.** O corpo contém várias partes independentes. O cabeçalho multipart precisa definir a fronteira, um limite, entre cada parte. A fronteira é utilizada com um parâmetro. Ela é uma *string* sinalizada que é repetida antes de cada parte em uma linha separada, precedida por dois hifens. O corpo será finalizado usando a sinalização de fronteira precedida e sucedida por dois hifens.

Quatro *subtypes* estão definidos para este *type*: *mixed*, *parallel*, *digest* e *alternative*. No *subtype mixed*, as partes devem ser apresentadas ao receptor na ordem exata, como na mensagem. Cada parte possui um tipo diferente e é definida na fronteira. O *subtype parallel* é similar ao *subtype mixed*, exceto que a ordem das partes não é importante. O *subtype digest* também é similar ao *subtype mixed*, exceto que o padrão *type/subtype* torna-se message/RFC822, como definido abaixo. No *subtype alternative*, a mesma mensagem é repetida usando diferentes formatos. A seguir temos um exemplo de mensagem multipartes usando o *subtype mixed*:

```
Content-Type: multipart/mixed; boundary=xxxx
--xxxx
Content-type: text/plain
-----
--xxxx
Content-type: image/gif
-----
--xxxx--
```

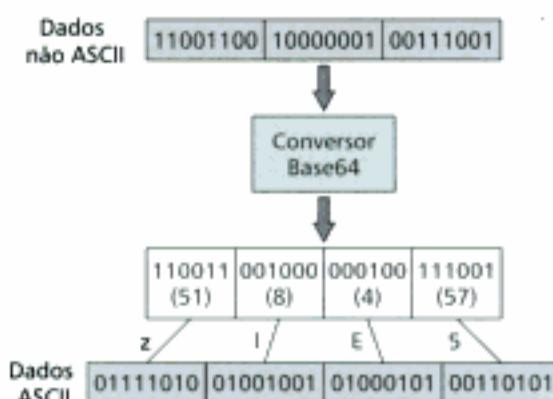
- **Message.** No *type* mensagem, o corpo como um todo é um mail, uma parte de um mail ou um ponteiro para uma mensagem. Três tipos são usados atualmente: *RFC822*, *partial* ou *external-body*. O *subtype RFC822* é usado se o corpo foi encapsulado noutra mensagem (incluindo o cabeçalho e corpo). O *subtype partial* é usado se a mensagem original foi fragmentada em diferentes mensagens e a mensagem que está sendo tratada é um dos fragmentos. Os fragmentos devem ser remontados no destino pelo MIME. Nesse caso, são necessários três parâmetros: *id*, *number* e *total*. O parâmetro *id* identifica a mensagem e se faz presente em todos os fragmentos. O parâmetro *number* define a ordem de seqüência do fragmento. O parâmetro *total* define a quantidade de fragmentos contidos na mensagem original. A seguir temos um exemplo de mensagem composta de três fragmentos:

```
Content-Type: message/partial;
id="forouzan@challenger.atc.fhda.edu";
number=1;
total=3;
```

O *subtype external-body* indica que o corpo não contém a mensagem real, mas somente uma referência à mensagem original (um ponteiro). Os parâmetros seguintes ao *subtype* definem o modo de acessar a mensagem original. Abaixo temos um exemplo:

```
Content-Type: message/external-body;
name="report.txt";
site ="fhda.edu";
access-type="ftp";
```

Hidden page

**Figura 26.6** Base64.

Toda seção de 6-bits é interpretada como um caractere de acordo com a Tabela 26.3.

TABELA 26.3 Tabela de codificação base64

Valor	Código										
0	A	11	L	22	W	33	h	44	s	55	3
1	B	12	M	23	X	34	i	45	t	56	4
2	C	13	N	24	Y	35	j	46	u	57	5
3	D	14	O	25	Z	36	k	47	v	58	6
4	E	15	P	26	a	37	l	48	w	59	7
5	F	16	Q	27	b	38	m	49	x	60	8
6	G	17	R	28	c	39	n	50	y	61	9
7	H	18	S	29	d	40	o	51	z	62	+
8	I	19	T	30	e	41	p	52	0	63	/
9	J	20	U	31	f	42	q	53	1		
10	K	21	V	32	g	43	r	54	2		

- **Quoted-printable.** O esquema de codificação Base64 é redundante, isto é, 24-bits são convertidos em quatro caracteres e eventualmente são transmitidos como 32-bits. O overhead nesse tipo de transmissão é 33,3%. Se a massa de dados consistir na maior parte de caracteres ASCII, com uma pequena porção de caracteres não ASCII, podemos usar a codificação quoted-printable. Se um caractere é ASCII, ele é enviado como tal. Caso não seja, ele é enviado como três caracteres. O primeiro caractere representa o sinal de igualdade (=). Os próximos dois caracteres são a representação hexadecimal do byte. A Figura 26.7 mostra um exemplo.

Content-Id

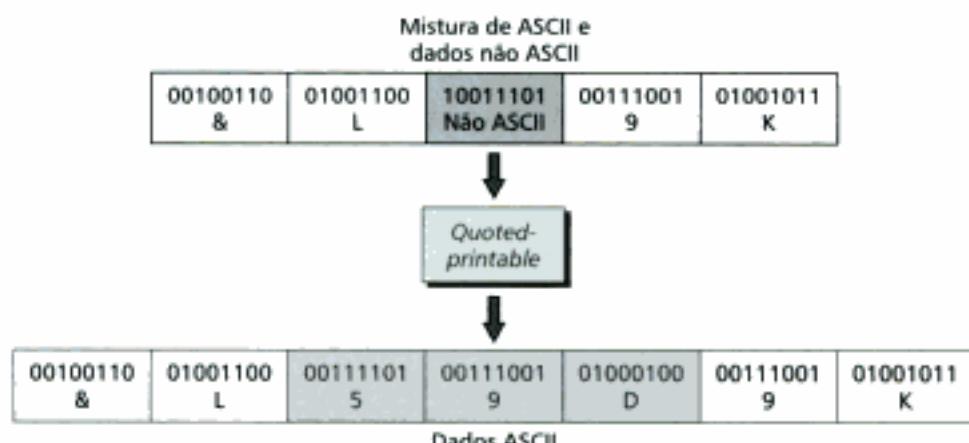
Este cabeçalho identifica unicamente toda a mensagem em um ambiente de múltiplas mensagens.

`Content-Id: id=<content-id>`

Content-Description

Este cabeçalho define se o corpo da mensagem é uma imagem, áudio ou vídeo.

`Content-Description: <description>`

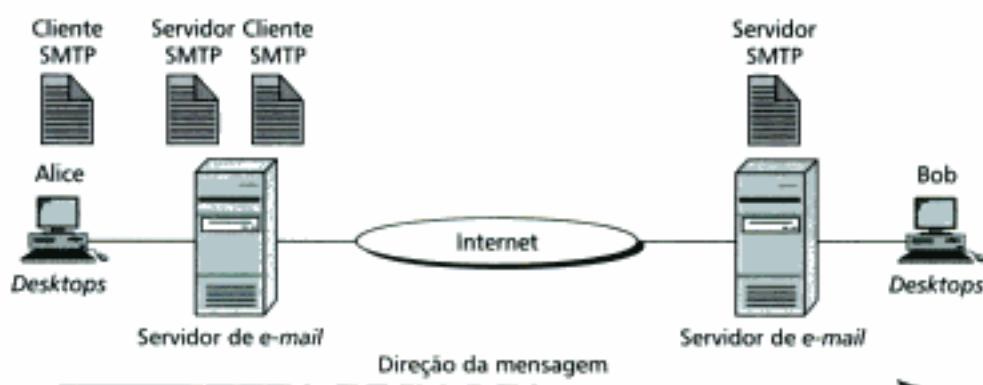
**Figura 26.7** Quoted-printable.

Mail Transfer Agent (MTA)

A transferência de *e-mail* real é feita através dos **Mail Transfer Agents (MTAs)**. Para transmitir um *e-mail*, um sistema deve possuir um cliente MTA e, para receber um *e-mail*, um sistema deve possuir um servidor MTA.

Na Internet, vimos que a transferência de mensagem é feita através do protocolo SMTP. Para transmitir uma mensagem precisamos de um cliente SMTP e um servidor SMTP. Na Figura 26.8, mostramos Alice enviando um *e-mail* ao Bob, com os clientes e servidores SMTP sendo apontados ao longo do caminho.

Observe que a transferência da mensagem ocorre entre dois servidores de *e-mail*, um no lado onde está Alice e outro no lado onde está Bob. Esses servidores de *e-mail* podem pertencer aos ISPs (provedor de Internet) onde os dois são assinantes ou às empresas onde eles trabalham.

**Figura 26.8** Cliente e servidor MTA.

Comandos e Respostas

O SMTP usa comandos e respostas para transferir mensagens entre um cliente MTA e um servidor MTA (veja a Figura 26.9). Cada comando ou resposta termina por um caractere duplo simbólico no fim da linha.

**Figura 26.9** Comandos e respostas.

Hidden page

Hidden page

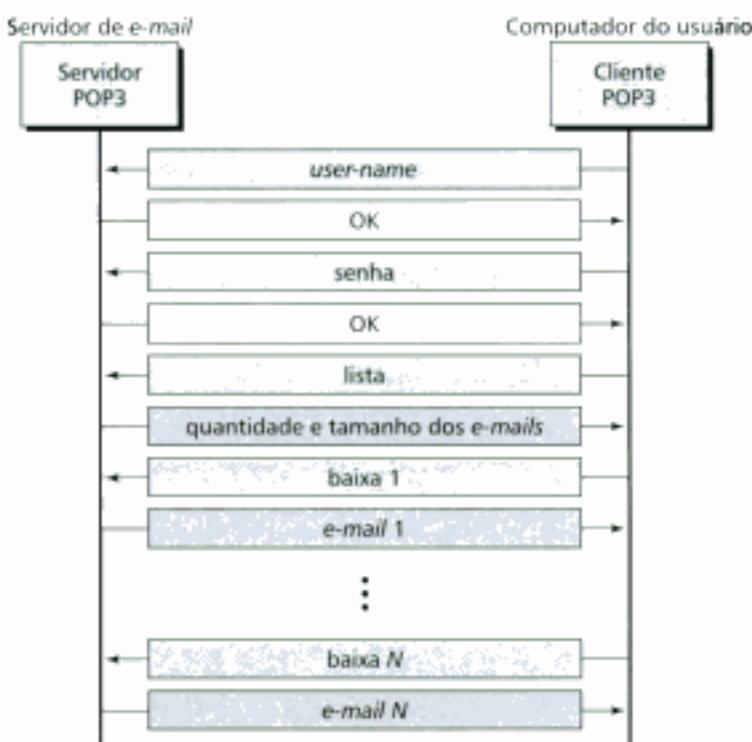


Figura 26.11 POP3.

IMAP4

O POP3 assume que toda vez que um cliente acessa o servidor toda a caixa de correio será transferida. Isto não é conveniente para as pessoas que acessam as caixas de correio de diferentes clientes (em casa, no trabalho, no hotel, etc.).

O POP3 é deficiente nesses casos. Ele não permite que o usuário organize *e-mails* no próprio servidor; além disso, o POP3 não permite que o usuário tenha pastas (*folders*) diferentes no servidor (claro que o usuário pode criar pastas no seu próprio computador). Por último, o POP3 não permite que um usuário verifique parcialmente o conteúdo da caixa antes de iniciar o *downloading*.

O **IMAP4 (Internet Mail Access Protocol – versão 4)** é outro protocolo de acesso à caixa similar ao POP3, mas com muitos recursos adicionais, pois o IMAP4 é bem mais complexo e muito mais poderoso.

O IMAP4 oferece as seguintes funções extra:

- Um usuário pode verificar os *e-mails* antes de baixá-los.
- Um usuário pode fazer uma busca de um determinado conteúdo em um *e-mail* antes de baixá-lo.
- Um usuário pode baixar apenas as mensagens selecionadas. Isto é especialmente útil quando a largura de banda do *link* utilizado é baixa e o *e-mail* contém "arquivos pesados", como de multimídia.
- Um usuário pode criar, apagar ou renomear as caixas de correio no servidor de *e-mail*.
- Um usuário pode criar uma hierarquia de caixas de correio em uma pasta (*folder*) para armazenamento de *e-mail*.

E-mail Baseado na Web

O *e-mail* tornou-se um recurso tão comum para as pessoas que hoje inúmeros *websites* oferecem esse serviço gratuitamente para qualquer pessoa que tiver um computador conectado à Internet. Dois *sites* bastante comuns são o Yahoo e o Hotmail. A idéia é muito simples. Os *e-mails* são transferidos, através de HTTP (veja Capítulo 27), do *browser* de Internet para o servidor de *e-mail* de um

Hidden page

A conexão de controle é mantida durante toda interação entre sessões FTP. A transferência de dados é aberta e, em seguida, fechada para cada arquivo transferido. Ela é aberta toda vez que são usados comandos envolvidos na transferência de arquivos e é fechada quando o arquivo é transferido. Em outras palavras, quando um usuário inicia uma sessão FTP, a conexão de controle é aberta. Enquanto a conexão de controle estiver aberta, a conexão de dados pode ser aberta e fechada diversas vezes, para transferência múltipla de arquivos.

Conexão

As duas conexões FTP, controle e dados, usam estratégias e números de portas diferentes.

Conexão de Controle

A **conexão de controle** é criada do mesmo modo que outros programas aplicativos descritos antes. A conexão permanece aberta durante todo o processo. O tipo de serviço usado pelo IP nessa conexão é o *minimize delay* (minimiza atrasos), porque esta é uma conexão interativa entre um usuário (pessoa) e um servidor. O usuário digita os comandos e espera que a resposta chegue, sem atrasos significativos.

Conexão de Dados

A **conexão de dados** usa a porta número 20 no lado de servidor. A conexão de dados é aberta quando dados estão prontos para serem transferidos e é fechada quando ela não é mais necessária. Uma conexão de dados pode ser aberta e fechada inúmeras vezes durante uma sessão; ao contrário da conexão que é aberto uma única vez. O tipo de serviço usado pelo IP nessa conexão é o *maximize throughput* (maximiza o throughput).

Comunicação

O cliente e o servidor FTP, os quais rodam em computadores diferentes, devem se comunicar. Estes dois computadores usam sistemas operacionais diferentes; o mesmo ocorrendo com conjunto de caracteres, estrutura e formato de arquivo que, em geral, são diferentes. O FTP deve fazer a compatibilidade entre todas essas heterogeneidades.

O FTP possui duas abordagens, uma para a conexão de controle e a outra para a conexão de dados. Estudaremos cada abordagem separadamente.

Comunicação sobre a Conexão de Controle

O FTP usa a mesma abordagem que o SMTP para se comunicar através da conexão de controle. O FTP usa o conjunto de caracteres ASCII (veja Figura 26.13). A comunicação acontece através de uma sequência de comandos e respostas. Este método simples é adequado para a conexão de controle porque podemos enviar um comando (resposta) por vez. Todo comando ou resposta ocupa uma única linha (curta), então não precisamos nos preocupar com o formato ou estrutura do arquivo. Cada linha é encerrada com dois caracteres (*carriage return* e *line feed*) no final.



Figura 26.13 Usando a conexão de controle.

Comunicação sobre a Conexão de Dados

O propósito e a implementação da conexão de dados são diferentes daqueles abordados na conexão de controle. O fato básico é: queremos transferir arquivos através da conexão de dados. O cliente deve definir o tipo de arquivo a ser transferido, a estrutura dos dados e o modo de transmissão. Além disso, a transmissão deve ser preparada pela conexão de controle antes que um arquivo possa ser transmitido através da conexão de dados. O problema da heterogeneidade é resolvido definindo três atributos de comunicação: tipo de arquivo, estrutura de dados e modo de transmissão (veja a Figura 26.14).



Figura 26.14 Usando a conexão de dados.

Tipo de Arquivo O FTP pode transferir, através da conexão de dados, um dos seguintes tipos de arquivos:

- **ASCII.** Este é o formato padrão para transferência de arquivos texto. Cada caractere é codificado usando ASCII. Assim, o transmissor converte os arquivos do formato original em caracteres ASCII e o receptor converte novamente os caracteres ASCII para o formato original.
- **EBCDIC.** Se uma ou as duas extremidades da conexão usarem a codificação EBCDIC (usada nos *mainframes* IBM) o arquivo poderá ser transferido usando essa codificação.
- **Imagen.** Este é o formato padrão para transferência de arquivos binários. O arquivo é enviado como um fluxo contínuo de *bits* sem qualquer tipo de interpretação ou codificação. Este formato é usado principalmente para transferir arquivos binários, tal como programas compilados ou imagens codificadas em 0s e 1s.

Se o arquivo estiver codificado em ASCII ou EBCDIC, outro atributo deve ser adicionado para definir a qualidade de impressão do arquivo.

1. **Nonprint.** Este é o formato padrão para transferência de um arquivo texto. O arquivo não contém especificações verticais para impressão. Isto significa que o arquivo não pode ser impresso sem um processamento adicional porque não há caracteres inteligíveis para serem interpretados pelo movimento vertical do mecanismo de impressão. Este formato é usado pelos arquivos que serão armazenados e processados no futuro.
2. **TELNET.** Neste formato, o arquivo contém caracteres ASCII verticais tal como o CR (*Carriage Return*), LF (*Line Feed*), NL (*New Line*) e o VT (*Vertical Tab*). O arquivo é imprimível após a transferência.

Estrutura de Dados O FTP pode transferir arquivos através da conexão de dados usando uma das seguintes interpretações da estrutura dos dados:

- **Arquivos (padrão).** O arquivo não tem estrutura. Ele é transmitido como um fluxo contínuo de *bytes*.

- **Registros.** O arquivo é dividido em registros ou estruturas em C. Esse tipo pode ser usado somente com arquivos texto.
- **Páginas.** O arquivo é dividido em páginas, cada qual devidamente numerada e identificada com cabeçalho. As páginas podem ser armazenadas ou acessadas, aleatória ou seqüencialmente.

Modo de Transmissão O FTP pode transferir um arquivo através da conexão de dados usando um dos seguintes modos de transmissão:

- **Cadeias.** Este é o modo padrão. Os dados são entregues do FTP para o TCP como uma cadeia contínua de *bytes*. O TCP é responsável pela separação dos dados em segmentos de tamanho apropriado. Se os dados são simplesmente uma cadeia de *bytes* (estrutura de arquivo), não é necessária identificação de fim de linha. Nesse caso, a indicação de fim de linha é o fechamento da conexão de dados pelo transmissor. Se os dados são divididos em registros, cada registro será acompanhado do caractere EOR (*End-Of-Record*) e o fim do arquivo terá o EOF (*End-Of-File*), ambos de 1 byte de tamanho.
- **Bloco.** Dados podem ser entregues do FTP ao TCP em blocos. Nesse caso, cada bloco é precedido de um cabeçalho de 3 *bytes*. O primeiro *byte* é denominado *descriptor block* (bloco descritor); os outros dois *bytes* definem o tamanho do bloco, em *bytes*.
- **Compressão.** Se o arquivo é muito grande, os dados podem ser comprimidos antes de serem enviados. O método de compressão usado normalmente toma unidade de dados que aparecem consecutivamente e os substituem por uma única ocorrência, seguido do número de repetições. Em um arquivo texto, há muitos espaços vazios. Em um arquivo binário, caracteres nulos são normalmente comprimidos.

Transferência do Arquivo

A transferência do arquivo ocorre na fase da conexão de dados, sobre o controle dos comandos enviados pela conexão de controle. Entretanto, lembre-se que a transferência de arquivos no FTP significa uma das três opções (veja a Figura 26.15).

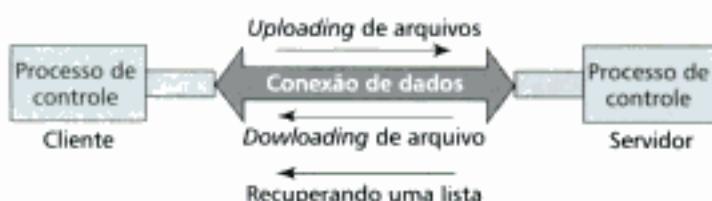


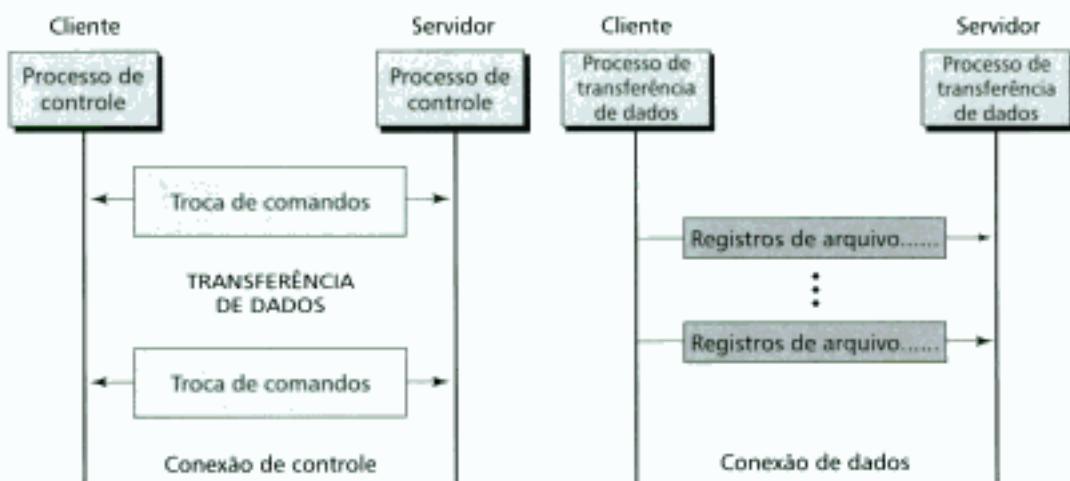
Figura 26.15 Transferência de arquivo.

- Um arquivo é baixado do servidor para o cliente. Isto é denominado *downloading de arquivo*.
- Um arquivo do cliente é copiado no servidor. Isto é denominado *uploading de arquivo*.
- Uma lista de diretório ou nomes de arquivos é enviada do servidor para o cliente. Observe que o FTP trata uma lista de diretório ou nomes de arquivos como um arquivo comum. Ele é enviado sobre a conexão de dados.

Exemplo 1

A Figura 26.16 mostra um exemplo de armazenamento de arquivo.

1. A conexão de controle é criada; muitos comandos e respostas de controle são trocados.
2. Os dados são transferidos, registro por registro.
3. Poucos comandos e respostas são trocados para fechamento da conexão.

**Figura 26.16** Exemplo 1.

Interface do Usuário

A maioria dos sistemas operacionais oferecem uma interface do usuário para acessar os serviços de FTP. A interface lembra ao usuário que um comando de entrada apropriado é esperado. Após o usuário digitar uma linha, a interface do FTP lê o comando na linha e o modifica para o comando FTP correspondente. A Tabela 26.4 mostra os comandos da interface do usuário FTP UNIX. Alguns comandos podem ser abreviados, contanto que não provoquem ambigüidade.

TABELA 26.4 Lista de comandos FTP no UNIX

Comandos
!, \$, account, append, ascii, bell, binary, bye, case, cd, cdup, close, ct, delete, debug, dir, disconnect, form, get, glob, hash, help, lcd, ls, macdef, mdelete, mdir, mget, mkdir, mls, mode, mput, nmap, ntrans, open, prompt, proxy, sendport, put, pwd, quit, quote, recv, remotehelp, rename, reset, rmdir, runique, send, status, struct, sunique, tenex, trace, type, user, verbose, ?

Exemplo 2

Mostramos alguns dos comandos da interface do usuário que realizam a tarefa proposta no Exemplo 1. As entradas realizadas pelo usuário são mostradas em negrito. Como pode ser visto abaixo, alguns dos comandos são fornecidos automaticamente pela interface. O usuário recebe um *prompt* e entra somente com os argumentos.

```

$ ftp challenger.atc.fhda.edu
Connected to challenger.atc.fhda.edu
220 Server ready
Name: forouzan
Password: xxxxxxxx
ftp > ls /usr/user/report
200 OK
150 Opening ASCII mode
.....
.....
226 transfer complete
ftp > close
221 Goodbye
ftp > quit
  
```

Hidden page

Hidden page

19. O campo _____ no cabeçalho MIME contém o tipo de dados e o corpo da mensagem.
- Content-type*
 - Content-transfer-encoding*
 - Content-id*
 - Content-description*
20. O campo _____ no cabeçalho MIME usa texto para descrever os dados no corpo da mensagem.
- Content-type*
 - Content-transfer-encoding*
 - Content-id*
 - Content-description*
21. O campo _____ no cabeçalho MIME descreve o método usado para codificar os dados.
- Content-type*
 - Content-transfer-encoding*
 - Content-id*
 - Content-description*
22. O campo _____ no cabeçalho MIME contém o tipo (*type*) e subtipo (*subtype*) dos dados.
- Content-type*
 - Content-transfer-encoding*
 - Content-id*
 - Content-description*
23. Uma imagem JPEG é enviada por *e-mail*. Qual é o *content-type*?
- Multipart/mixed*
 - Multipart/image*
 - Image/JPEG*
 - Image/basic*
24. Um *e-mail* contém uma saudação de aniversário textual, um figura de um bolo e uma canção. O texto deve prececer a imagem. Qual é o *content-type*?
- Multipart/mixed*
 - Multipart/parallel*
 - Multipart/digest*
 - Multipart/alternative*
25. Um *e-mail* contém uma saudação de aniversário textual, um figura de um bolo e uma canção. A ordem não é importante. Qual é o *content-type*?
- Multipart/mixed*
 - Multipart/parallel*
 - Multipart/digest*
 - Multipart/alternative*
26. Uma mensagem é fragmentada em três mensagens de *e-mail*. Qual é o *content-type*?
- Multipart/mixed*
 - Multipart/parallel*
 - Multipart/digest*
 - Multipart/alternative*
27. Uma máquina cliente usualmente precisa _____ para enviar um *e-mail*.
- Somente do SMTP
 - Somente POP
 - Tanto do SMTP quanto POP
 - Nenhuma das alternativas anteriores
28. Qual das afirmações a seguir é a verdadeira?
- FTP permite a transferência de arquivos entre sistemas com diferentes estruturas de diretório
 - FTP permite a transferência de arquivos entre um sistema usando ASCII e outro sistema, usando o EBCDIC
 - FTP permite a transferência de arquivos entre um PC e uma estação SUN
 - Todas as alternativas anteriores estão corretas
29. Durante uma sessão FTP, a conexão de controle é aberta _____.
- Uma única vez
 - Duas vezes
 - Tantas vezes quanto se fizerem necessárias
 - Todas as alternativas anteriores
30. Durante uma sessão FTP, a conexão de dados é aberta _____.
- Uma única vez
 - Duas vezes
 - Tantas vezes quanto se fizerem necessárias
 - Todas as alternativas anteriores
31. No FTP, quais atributos devem ser definidos pelo cliente antes da transmissão ser estabelecida?
- Tipos de dados
 - Estrutura de arquivos
 - Modo de transmissão
 - Todas as alternativas anteriores

Hidden page

Hidden page

HTTP e WWW

A World Wide Web (WWW) modificou nosso modo de viver. As pessoas tem tomado cada vez mais consciência do poder da Internet e, em particular, da vertente WWW. Neste capítulo, discutiremos inicialmente o protocolo HTTP, um protocolo de transferência de arquivos especialmente projetado para facilitar o acesso à WWW. Em seguida, discutiremos a WWW propriamente dita.

27.1 HTTP

O **HyperText Transfer Protocol (HTTP)** é usado principalmente para acessar dados na World Wide Web. O protocolo permite a transferência de dados na forma de textos simples, hipertexto, áudio, vídeo e muitas outras formas. Ele é denominado protocolo de transferência de hipertexto porque é usado em um ambiente onde há transições rápidas de um documento para outro.

O HTTP funciona como uma combinação do FTP com o SMTP. Ele se parece com o FTP porque permite a transferência de arquivos e usa os serviços do TCP. Entretanto, as semelhanças param por aí, porque ele é muito mais simples que o FTP e usa apenas uma conexão TCP (a porta número 80). Não há nenhum controle separado da conexão; apenas dados são transferidos entre o cliente e o servidor.

O HTTP é parecido com o SMTP no sentido que os dados são transferidos entre o cliente e o servidor como mensagens semelhantes àquelas encontradas no SMTP. Além disso, o formato das mensagens é controlado por cabeçalhos muito parecidos com o cabeçalho MIME. Entretanto, o HTTP difere do SMTP no modo como as mensagens são enviadas do cliente ao servidor. Diferentemente das mensagens SMTP, as mensagens HTTP não se destinam à leitura direta por pessoas. Elas são lidas e interpretadas pelo servidor HTTP e o cliente HTTP (*browser*). As mensagens SMTP são armazenadas e encaminhadas, já as mensagens HTTP são transmitidas imediatamente.

A idéia do HTTP é muito simples. Um cliente envia um pedido, na forma de uma mensagem, ao servidor. O servidor envia a resposta, também na forma de uma mensagem, ao cliente. As mensagens pedido e resposta transportam dados na forma de documento com um formato bastante parecido ao MIME.

Os comandos do cliente ao servidor são inseridos numa mensagem de solicitação, tipicamente um documento (doc.). Os itens do arquivo solicitado ou outras informações são inseridas numa mensagem de resposta também semelhante a um doc.

O protocolo de aplicação HTTP usa os serviços do protocolo de transporte TCP na porta número 80.

Transação HTTP

A Figura 27.1 ilustra uma transação HTTP entre um cliente e um servidor. Embora o HTTP use os serviços do protocolo TCP, que funciona baseado em um diagrama de estados, o HTTP é por si só um protocolo sem estado (*stateless*), isto é, não se “lembra” dos pedidos anteriores. O cliente inicia a transação enviando uma mensagem de solicitação (um pedido). Em seguida, o servidor responde enviando a mensagem resposta.



Figura 27.1 Transação HTTP.

Há dois tipos gerais de mensagens HTTP: pedido e resposta. Ambos tipos de mensagens seguem quase o mesmo formato.

Mensagem Pedido

Um pedido consiste da linha pedido, cabeçalhos e, às vezes, de um corpo. A Figura 27.2 ilustra o conceito.

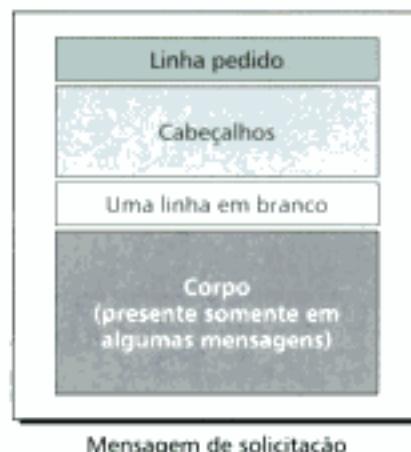


Figura 27.2 Mensagem de solicitação (pedido).

Linha Pedido

A **linha de solicitação (pedido)** define o tipo de pedido, recurso (URL) e a versão HTTP (veja a Figura 27.3).

- **Tipo de pedido.** Na versão 1.1 do HTTP, estão definidos muitos tipos de pedidos (mensagens de solicitação). O tipo de pedido classifica as mensagens de solicitação em diversos métodos (comandos), como veremos adiante.
- **Recurso (Uniform Resource Locator – URL).** Um cliente que deseja acessar uma página da Web precisa de um endereço. Para facilitar o acesso aos documentos distribuídos

**Figura 27.3** Linha pedido.

mundo afora, o HTTP usa o conceito de URL. O URL é um padrão para especificar qualquer tipo de informação na Internet. O URL define quatro partes: protocolo, *host*, porta e caminho (*path*) (veja a Figura 27.4).

**Figura 27.4** URL.

- O campo *protocolo* diz respeito ao protocolo de aplicação usado na obtenção do documento. Muitos protocolos diferentes podem ser utilizados na aquisição de um documento, entre eles citamos o FTP e o HTTP.
- O *host* é o computador onde a informação está localizada, embora o nome do computador possa ser um nome alternativo (alias). As páginas da Web normalmente ficam armazenadas em computadores e aos computadores são dados nomes alternativos que normalmente começam com os caracteres *www*. Isto não é obrigatório, um *host* que hospeda uma página da Web pode ter qualquer nome.
- O URL pode conter opcionalmente o número da porta do servidor. Se o número de porta estiver incluído, ele deve ser inserido entre o campo *host* e o caminho (*path*). Além disso, deve ser separado da parte de *host* por dois pontos.
- O campo **caminho (*path*)** indica como encontrar (o caminho) um arquivo onde a informação desejada está localizada. Observe que a informação de caminho pode conter barras, no formato padrão do sistema operacional UNIX, separando diretórios, subdiretórios e arquivos.
- **Versão.** Embora a versão mais atual do HTTP seja a 1.1, é possível encontrarmos as versões 1.0 e 0.9, pois elas ainda estão em uso em alguns sistemas.

Métodos (Comandos)

O campo tipo de pedido na mensagem de solicitação define diversos tipos de mensagens tratadas como *métodos* ou *comandos*. De fato, o método de solicitação é um comando real que um cliente envia a um servidor. A seguir, descreveremos em poucas palavras os propósitos de alguns desses comandos.

GET O comando GET é utilizado quando um cliente deseja obter um documento do servidor. O endereço do documento é definido no URL. Este é o principal método de obtenção de documentos. O servidor usualmente responde com os itens do documento no corpo da mensagem resposta, a menos que ocorra um erro.

HEAD O comando HEAD é usado quando o cliente deseja obter alguma informação sobre um documento e não o documento em si. Ele é semelhante ao GET, mas a resposta do servidor não contém um corpo.

POST O comando POST é usado pelo cliente para fornecer alguma informação ao servidor. Por exemplo, ele pode ser utilizado de modo a passar uma entrada a um servidor.

PUT O comando PUT é usado pelo cliente para fornecer um documento novo ou atualizado a ser armazenado no servidor. O documento é inserido no corpo do pedido e armazenado no local definido pelo URL.

PATCH O comando PATCH é semelhante ao PUT, exceto que o pedido contém uma lista de modificações que devem ser implementadas em um arquivo existente.

COPY O comando COPY, como sugere o nome, copia um arquivo para outra localização. A localização do arquivo de origem é fornecida na linha pedido (URL). Por outro lado, a localização do destino é dada na seção Cabeçalho da Mensagem (discutido mais adiante).

MOVE O comando MOVE, outra vez seguindo a sugestão do nome, move um arquivo para outra localização. A localização do arquivo de origem é fornecida na linha pedido (URL) e a localização do destino é dada na seção Cabeçalho da Mensagem.

DELETE O comando DELETE apaga um documento do servidor.

LINK O comando LINK estabelece um ou mais *links* de um documento para outra localização. A localização do arquivo é dada na linha pedido (URL), já a localização do destino é dada na seção Cabeçalho da Mensagem.

UNLINK O comando UNLINK apaga um ou mais *links* criados pelo comando LINK.

OPTION O comando OPTION é usado pelo cliente durante as perguntas ao servidor sobre as opções disponíveis.

Mensagem Resposta

Uma mensagem resposta consiste de uma linha de *status*, um cabeçalho e, às vezes, um corpo. A Figura 27.5 ilustra essas partes.

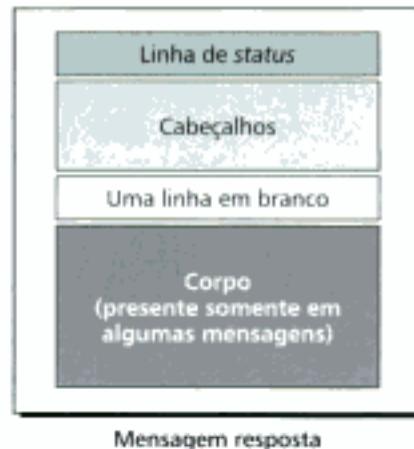


Figura 27.5 Mensagem resposta.

Linha de Status

A **linha de status** define o estado da mensagem resposta. Ela consiste da versão HTTP, um espaço, código de *status*, um espaço e uma frase de *status*. Veja a Figura 27.6.

Versão HTTP Este campo é o mesmo campo correspondente na linha pedido.

Código de Status Este campo é semelhante aos campos de código nos protocolos FTP e SMTP. Ele consiste de três dígitos.

Frase de Status Este campo explica o código de *status* na forma de texto.



Figura 27.6 Linha de status.

Cabeçalhos das Mensagens

Os cabeçalhos destinam-se às trocas de informações adicionais entre um cliente e um servidor. Por exemplo, um cliente solicitando um documento em um formato especial ou o servidor enviando informação extra sobre um documento específico.

O cabeçalho pode ser formado de uma ou mais linhas. Cada linha de cabeçalho é constituída do nome do cabeçalho, dois pontos, um espaço e um valor do cabeçalho (veja a Figura 27.7). Mostraremos algumas linhas de cabeçalho no final desta seção.

Uma linha de cabeçalho pertence a um dos quatro tipos: cabeçalho geral, cabeçalho de opções do cliente (pedido), cabeçalho resposta e cabeçalho entidade. Uma mensagem pedido pode conter apenas os cabeçalhos geral, opções do cliente e entidade. Por outro lado, uma mensagem de resposta pode conter somente os cabeçalhos geral, resposta e entidade. O diagrama da Figura 27.8 ilustra a estrutura dessas duas mensagens (pedido e resposta).



Figura 27.7 Formato do cabeçalho.



Figura 27.8 Cabeçalhos.

Cabeçalho Geral

O **cabeçalho geral** fornece as informações genéricas sobre uma mensagem e pode estar presente tanto em um pedido quanto em uma resposta.

Cabeçalho Opções do Cliente (Pedido)

O **cabeçalho opções do cliente (pedido)** pode estar presente apenas nas mensagens pedido. Ele especifica a configuração e as preferências de formato de documento do cliente.

Cabeçalho Resposta

O **cabeçalho resposta** pode estar presente apenas nas mensagens resposta. Ele especifica a configuração do servidor e as informações especiais sobre o pedido.

Cabeçalho Entidade

O **cabeçalho entidade** fornece informações sobre o corpo do documento. Embora esse cabeçalho esteja presente, na maioria das vezes, nas mensagens resposta, algumas mensagens pedido que contém um corpo, tal como os comandos POST e PUT, também usam este tipo de cabeçalho.

Alguns Exemplos

Nesta seção estudaremos dois exemplos de transação envolvendo cliente e servidor.

Exemplo 1

Este exemplo mostra a obtenção de um documento. Usamos o comando GET para obter uma imagem no caminho (*path*) /usr/bin/image1. A linha pedido mostra o comando (GET), o URL e a versão do protocolo HTTP (1.1). O cabeçalho possui duas linhas informando que o cliente aceita imagens no formato GIF e JPEG. A mensagem de solicitação (o pedido) não possui um corpo. A mensagem resposta mostra a linha de *status* e quatro linhas de cabeçalho. As linhas de cabeçalho definem a data, o servidor, a versão MIME e o tamanho do documento. O corpo do documento segue logo após o cabeçalho (veja Figura 27.9).



Figura 27.9 Exemplo 1.

Exemplo 2

Este exemplo mostra a obtenção de informação a respeito de um documento. Usamos o comando HEAD para obter informação sobre um documento HTML (veja a próxima seção). A linha pedido mostra o comando (HEAD), o URL e a versão HTTP (1.1). Por sua vez, o cabeçalho aparece em uma única linha mostrando que o cliente pode aceitar o documento em qualquer formato. Outra vez o pedido não possui um corpo. A mensagem resposta é composta da linha de *status* e de cinco linhas de cabeçalho. As linhas de cabeçalho definem a data, o servidor, a versão MIME, o tipo e o tamanho do documento (veja a Figura 27.10). Observe que a mensagem resposta também não possui um corpo.

Algumas Características Adicionais

Nesta seção descreveremos outras características do protocolo HTTP, versão 1.1.

Hidden page

correspondente na Web. Todas as respostas são recebidas, processadas e armazenadas para uso futuro de outros clientes, caso tenham sido aprovadas, pelo servidor de *proxy*.

Um servidor de *proxy* reduz a carga do servidor original, diminui o tráfego e melhora a latência da rede. Entretanto, os clientes devem estar configurados para acessar um servidor de *proxy* ao invés do servidor alvo.

27.2 WORLD WIDE WEB (WWW)

A **World Wide Web (WWW)** é um repositório de informação espalhado ao redor do mundo. A WWW combina flexibilidade, portabilidade e características amigáveis que a tornam distinta de outros serviços encontrados na Internet.

O projeto WWW foi iniciado pelos físicos do CERN (Centro Europeu de Pesquisas Nucleares) que necessitavam de um meio de viabilizar a distribuição de recursos indispensáveis aos cientistas espalhados por diversos países, através da troca de documentos.

Atualmente, a WWW é um serviço cliente-servidor distribuído, onde um cliente usando um *browser* pode acessar um serviço hospedado em um servidor. O serviço fornecido encontra-se distribuído em muitas localizações, chamadas *websites* (veja a Figura 27.11).

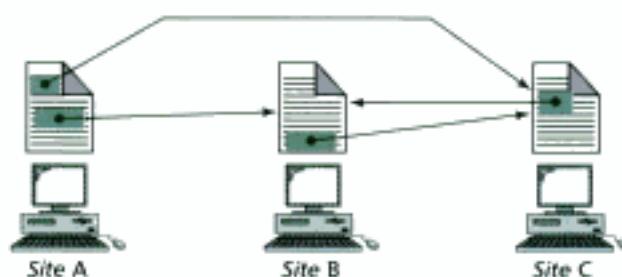


Figura 27.11 Serviços distribuídos.

Hipertexto e Hipermídia

A WWW funciona com base no conceito de hipertexto e hipermídia. Em um ambiente **hipertexto**, a informação é armazenada em um conjunto de documentos vinculados em um banco de dados através de referências (*hyperlinks*). Um item de um documento pode ser referenciado a outro documento através de um *hyperlink*. A pessoa que estiver pesquisando um documento pode alternar para outros documentos escolhendo (clicando) nos itens que fazem referência aos documentos desejados. A Figura 27.12 mostra o conceito de hipertexto.

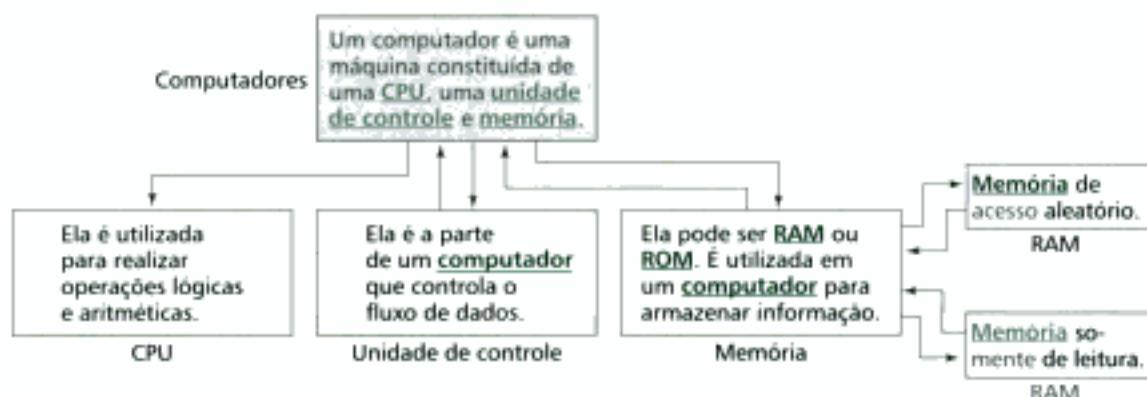


Figura 27.12 Hipertexto.

Considerando que os documentos de hipertexto contém somente texto, os documentos de **hipermídia** contém figuras, gráficos e som.

A unidade de hipertexto ou hipermídia disponível na Web é denominada **página**. A página principal de uma organização, empresa ou indivíduo recebe o nome de **homepage**.

A informação sobre um objeto específico pode estar concentrada ou distribuída. No primeiro caso, toda a informação pode consistir de uma ou mais páginas hospedadas em um mesmo servidor. No segundo caso, a informação está baseada em múltiplas páginas hospedadas em servidores distintos.

Arquitetura de um Browser

Existem diversas distribuições de *browsers* que interpretam e exibem um documento da Web e, praticamente, todos usam a mesma arquitetura. Usualmente, cada *browser* consiste de três partes: um controlador, programas cliente e interpretadores. O controlador recebe uma informação de um dispositivo de entrada (por exemplo, teclado ou *mouse*) e chama os programas cliente para acessar o documento. Logo após os documentos terem sido acessados, o controlador chama os serviços dos interpretadores para exibir o documento na tela. Os programas clientes podem ser um dos protocolos descritos anteriormente, tal como o HTTP, FTP ou SMTP. O interpretador pode se basear na linguagem HTML* ou Java, dependendo do tipo de documento (veja a Figura 27.13).

Um documento da Web pode ser classificado em uma das seguintes categorias: estático, dinâmico ou ativo (veja a Figura 27.14).

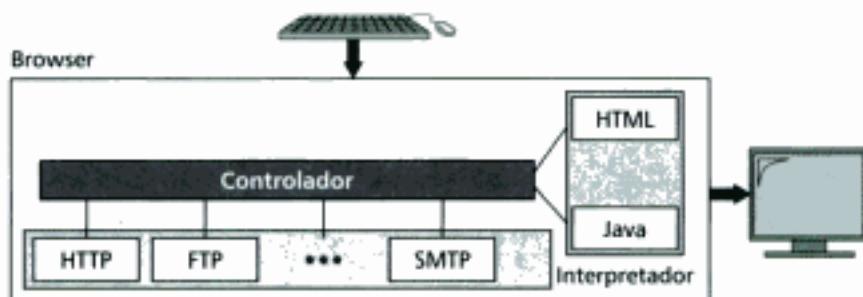


Figura 27.13 Arquitetura de um browser.

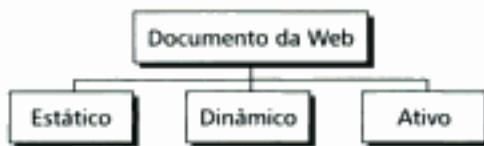


Figura 27.14 Tipos de documentos da Web.

Documentos Estáticos

Os **documentos estáticos** possuem estruturas fixas que são criadas e armazenadas em um servidor. O cliente tem permissão somente para realizar uma cópia do documento. Em outras palavras, o conteúdo estrutural do arquivo é determinado quando o arquivo é gerado e não quando é manipulado pelo cliente. Obviamente, os conteúdos armazenados em um servidor podem ser modificados, mas não são os usuários quem têm esse tipo de permissão. Quando um cliente acessa um documento, uma cópia do documento é enviada pelo servidor. O usuário pode usar um *browser* para exibir o documento na tela (veja a Figura 27.15).

* N. de R. T.: HTML não é uma linguagem de programação e não é voltada ao desenvolvimento de aplicações estruturadas ou orientadas a objetos.

Hidden page

Hidden page

TABELA 27.1 Algumas tags em HTML

<i>Tag inicial</i>	<i>Tag final</i>	<i>Descrição</i>
<i>Tags de esqueleto</i>		
<HTML>	</HTML>	Define um documento HTML
<HEAD>	</HEAD>	Define o cabeçalho do documento
<BODY>	</BODY>	Define o corpo do documento
<i>Tags de título e cabeçalho</i>		
<TITLE>	</TITLE>	Define o título do documento
<Hn>	</Hn>	Define cabeçalhos diferentes (n é um número inteiro)
<i>Tags de formatação de texto</i>		
		Negrito
<I>	</I>	Itálico
<U>	</U>	Sublinhado
_		Subscrito
[]	Sobrescrito
<i>Tags de fluxo de dados</i>		
<CENTER>	</CENTER>	Centrado
 	-	Quebra de linha
<i>Tags de lista</i>		
		Lista ordenada
		Lista não ordenada
		Um item de lista
<i>Tag de imagem</i>		
	-	Define uma imagem
<i>Tag de hyperlink</i>		
<A>		Define um endereço (<i>hyperlink</i>)
<i>Tag de conteúdo executável</i>		
<APPLET>	</APPLET>	O documento é um <i>applet</i>

Exemplo 3

Este exemplo mostra como as *tags* são usadas para permitir ao *browser* formatar a aparência do texto.

```

<HTML>
<HEAD>
  <TITLE> Primeiro exemplo de documento </TITLE>
</HEAD>
<BODY>
  <CENTER>
    <H1><B> ATENÇÃO </B></H1>
  </CENTER>
  Você pode obter um exemplar deste livro:
  <UL>
    <LI> Adquirindo-o junto à editora
    <LI> Solicitando-o on-line (no site)
    <LI> Comprando-o diretamente numa livraria
  </UL>
</BODY>
</HTML>

```

Hidden page



Figura 27.19 Documento dinâmico.

como os dados de entrada devem ser fornecidos ao programa e como o resultado de saída deve ser utilizado.

O CGI não é uma nova linguagem. Ao contrário, ele permite aos programadores usar qualquer ambiente de programação, como C, C++, Bourne Shell, Korn Shell, C Shell ou Perl. A única coisa que um CGI define é um conjunto de regras e termos que o programador deve seguir.

O termo *common* no CGI indica que o padrão define um conjunto de regras comuns à qualquer linguagem ou plataforma. O termo *gateway* significa que a aplicação CGI é uma porta (um tradutor) que pode ser utilizada para acessar outros recursos, tais como bancos de dados e pacotes gráficos. O termo *interface* sugere que há um conjunto de termos predefinidos, variáveis, chamadas, e assim por diante, que podem ser usados em qualquer aplicação CGI.

Aplicação CGI

Uma aplicação CGI é, na sua forma mais simples, um programa escrito em uma das linguagens que suportam CGI. Qualquer programador que consegue codificar uma seqüência lógica em um programa e conhece a sintaxe de uma das linguagens mencionadas acima pode escrever uma aplicação CGI simples.

Exemplos

Nesta seção daremos alguns exemplos de programas mostrando o conceito e a idéia por trás do CGI. Os exemplos de programas foram escritos em linguagens diferentes para mostrar ao leitor que o CGI independe da linguagem.

Exemplo 6

O Exemplo 6 é uma aplicação CGI contida em um *script* Bourne Shell. O programa acessa o utilitário UNIX *date* que retorna a data e a hora. Observe que a saída do programa é um texto simples.

```
#!/bin/sh
# O cabeçalho do programa
echo Content_type: text/plain
echo
# O corpo do programa
now='date'
echo $now
exit 0
```

Exemplo 7

O Exemplo 7 é bastante similar ao Exemplo 6, exceto que a saída do programa está em HTML.

```
#!/bin/sh
# O cabeçalho do programa
echo Content_type: text/html
echo
# O corpo do programa
echo <HTML>
echo <HEAD><TITLE> Data e Hora </TITLE></HEAD>
echo <BODY>
now='date'
echo <CENTER><B> $now </B></CENTER>
echo </BODY>
echo </HTML>
exit 0
```

Exemplo 8

O Exemplo 8 é similar ao Exemplo 7, exceto que o programa foi escrito na linguagem Perl.

```
#!/bin/perl
# O cabeçalho do programa
print "Content_type: text/html\n";
print "\n";
# O corpo do programa
print "<HTML>\n";
print "<HEAD><TITLE> Data e Hora </TITLE></HEAD>\n";
print "<BODY>\n";
$now = 'date';
print "<CENTER><B> $now </B></CENTER>\n";
print "</BODY>\n";
print "</HTML>\n";
exit 0
```

Documentos Ativos

Para muitas aplicações, precisaremos de um programa rodando no *site* do cliente. Estes são denominados **documentos ativos**. Por exemplo, imagine que desejamos rodar um programa que crie gráficos animados na tela interagindo com o usuário. O programa necessita definitivamente ser rodado no cliente onde a animação e a interação irão tomar lugar. Quando um *browser* solicita um documento ativo, o servidor envia uma cópia do documento codificado em *bytes*. Em seguida, o documento é rodado no *browser* do cliente (veja a Figura 27.20).

Um documento ativo no servidor é armazenado na forma binária. Entretanto, ele não gera tanto *overhead* para o servidor como documento dinâmico. Embora um documento ativo não rode no servidor, ele é armazenado como um documento binário que é obtido pelos clientes. Quando um cliente recebe um documento, ele também pode armazená-lo na área de armazenamento dele. dessa forma, o cliente pode rodar o documento tantas vezes quanto se fizerem necessárias, sem ter que solicitá-lo ao servidor.

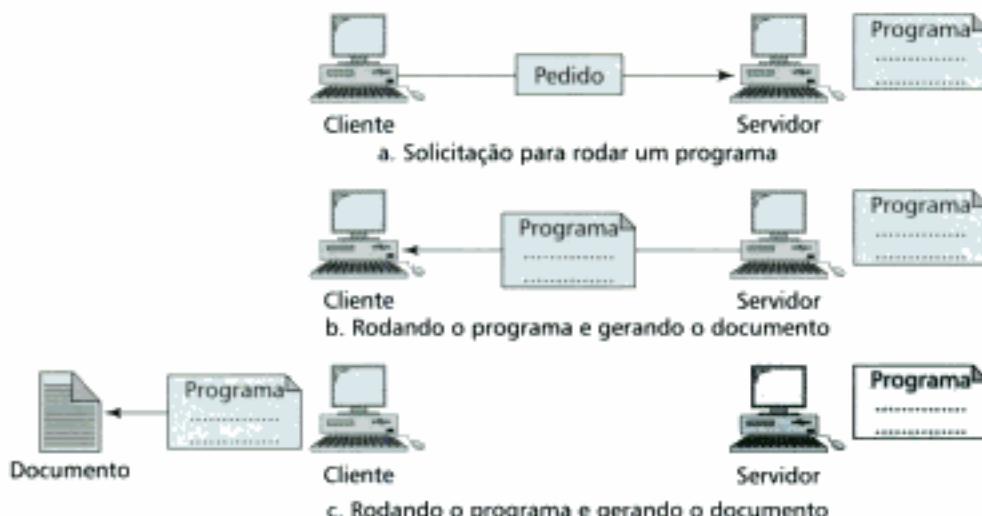


Figura 27.20 Documento ativo.

Um documento ativo é transportado do servidor para o cliente na forma binária. Isto significa que o servidor pode fazer compressão de dados e o cliente fazer descompressão quando tiver que usar o documento, preservando tanto largura de banda quanto tempo de transmissão.

Criação, Compilação e Execução

As etapas a seguir mostram como criar, compilar e executar documentos ativos.

1. No servidor, um programador escreve um programa, em código fonte, e o armazena num arquivo.
2. Ainda no servidor, o programa é compilado e o código binário é gerado, o qual é armazenado em um arquivo. A informação de caminho do arquivo é a mesma usada pelo URL para fazer referência ao arquivo. Neste arquivo, cada comando do programa (sentença) está na forma binária e cada identificador (variável, constante, nomes de função e assim por diante) é referido por um endereço binário equivalente.
3. Um cliente (*browser*) solicita uma cópia do código binário que é transportado, provavelmente, usando compressão de dados do servidor ao cliente (*browser*).
4. O cliente (*browser*) usa seu próprio *software* para transformar o código binário em código executável. O *software* vincula todas as bibliotecas e as torna prontas para execução.
5. O cliente (*browser*) roda o programa e gera o resultado que pode incluir animação ou interação com o usuário.

Java

A linguagem **Java** é uma combinação de programação de alto nível, alta *performance* (tempo de execução) e uma classe de bibliotecas que permitem a um programador escrever um documento ativo (um *applet*) e a um *browser* rodá-lo. Programas em Java também podem ser independentes (autônomos), sem a necessidade de um *browser*.

Java é uma linguagem orientada a objeto que é, sintáticamente e semanticamente, muito parecida com a linguagem C++. Entretanto, ela não carrega algumas das complexidades do C++, tais como a sobrecarga de operadores ou a herança múltipla. Além disso, Java é uma plataforma independente e não usa ponteiro aritmético. Na linguagem Java, como em outras linguagens orientadas a objetos, o programador define um conjunto de objetos e um conjunto de operações (métodos) para agir sobre esse objetos. Finalmente, Java é uma linguagem *concorrente*, o que significa que o programador pode usar cadeias múltiplas para criar concorrência.

Classes e Objetos

A linguagem Java, como muitas linguagens orientadas a objeto, usa o conceito de classes e objetos. Um objeto é uma instância de uma classe que usa métodos (procedimentos ou funções) para manipular dados encapsulados.

Herança

Uma das idéias principais da programação orientada a objetos é o conceito de herança. A herança define uma estrutura hierárquica de objetos, na qual um objeto pode herdar dados e métodos de outros objetos. Na linguagem Java podemos definir uma classe como uma classe base que contém dados e métodos comuns a muitas classes. As classes herdadas podem herdar estes dados e métodos e também podem possuir dados e métodos próprios.

Pacotes

A linguagem Java possui bibliotecas de classes bastante ricas, as quais permitem ao programador criar e manipular diferentes objetos em um *applet*.

Esqueleto de um Applet

Um *applet* é um documento ativo escrito em Java. Atualmente, um *applet* é a definição de uma classe herdada publicamente, a qual é herdada da classe *applet* definida na biblioteca `java.applet`. O programador pode definir dados privados e métodos públicos e privados (veja a Figura 27.21).

O processo cliente (*browser*) cria uma instância deste *applet*. Então, o *browser* usa os métodos públicos definidos no *applet* para invocar métodos privados ou para acessar os dados. A Figura 27.22 ilustra este relacionamento.

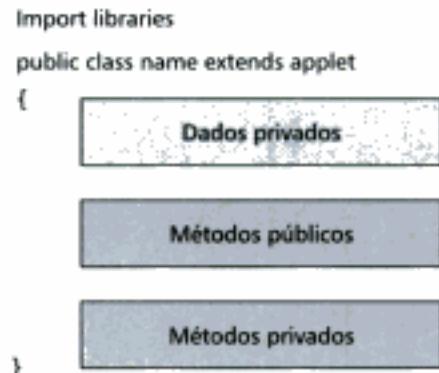


Figura 27.21 Esqueleto de um applet.

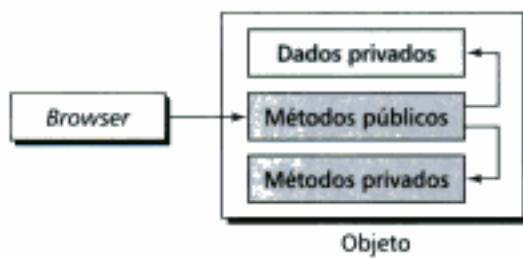


Figura 27.22 Objeto definido por um applet.

Criação, Compilação e Execução

A primeira etapa é usar um editor para criar um arquivo fonte em Java. O nome do arquivo é o mesmo nome da classe herdada publicamente, com a extensão ".java". A próxima etapa é o compilador codificar este arquivo em *bytes*, com a extensão ".class". A última etapa consiste em criar um *applet* que pode ser executado em um *browser* (veja a Figura 27.23).



Figura 27.23 Criação e compilação.

Documento HTML

Para usar o *applet* é gerado um documento HTML e o nome do *applet* é inserido entre os tags <APPLET>. O *tag* também define o tamanho da janela usada pelo *applet* (veja a Figura 27.24).

```

<HTML>
    <APPLET CODE = "Name.class"
    WIDTH = mmm
    HEIGHT = nnn .... >
    </APPLET >
</HTML>

```

Figura 27.24 Documento HTML transportando um *applet*.

Exemplos

Nesta seção daremos dois exemplos muito simples de programas em Java. O propósito não é ensinar Java, mas mostrar como a linguagem Java pode ser utilizada para criar documentos ativos.

Exemplo 9

Neste exemplo importaremos inicialmente dois pacotes (`java.awt` e `java.applet`). Eles contêm as declarações e definições das classes e dos métodos que necessitaremos. Nossa exemplo usa somente uma classe herdada publicamente, denominada *First*. Definiremos somente um método público, chamado *paint*. O *browser* pode acessar a instância da classe *First* através do método público *paint*. Porém, o método *paint* chama outro método, denominado *drawString*, o qual está definido no `java.awt.*`. Três parâmetros são passados ao método *drawString*: uma *string* que desejamos exibir, a coordenada *x* e a coordenada *y*. As coordenadas estão referenciadas ao topo esquerdo da janela do *browser* em *pixels*.

Hidden page

Hidden page

Hidden page

- c. Hipertexto, hipermídia HTML
d. Todas as alternativas anteriores
31. Que tipo de documento da Web é executado no cliente?
a. Estático
b. Dinâmico
c. Ativo
d. Todas as alternativas anteriores
32. Que tipo de documento da Web é criado no servidor somente quando é feito um pedido de um cliente?
a. Estático
b. Dinâmico
c. Ativo
d. Todas as alternativas anteriores
33. Que tipo de documento da Web possui estrutura fixa e é criado e armazenado no servidor?
a. Estático
b. Dinâmico
c. Ativo
d. Todas as alternativas anteriores
34. O _____ de uma página da Web contém o título e os parâmetros usados pelo *browser*.
a. Tag
b. Cabeçalho
c. Corpo
d. Atributo
35. Na linha ALIGN é _____.
a. Um tag
b. O cabeçalho
c. O corpo
d. Um atributo
36. Um tag final geralmente tem a seguinte forma _____.
a. </nome_tag>
b. <\nome_tag>
c. <nome_tag>
d. <nome_tag!>
37. Que categoria de tags HTML permite listar documentos?
a. Imagem
b. Lista
c. *Hyperlink*
d. Conteúdo executável
38. Os tags de _____ encerram códigos binários ou bytes de código.
a. Imagem
- b. Lista
c. *Hyperlink*
d. Conteúdo executável
39. Um programa pode usar _____ para escrever uma aplicação CGI.
a. Bourne shell script
b. Perl
c. C
d. Todas as alternativas anteriores
40. Um ator desempregado colocou o currículo dele na Web. Este é provavelmente um documento _____.
a. Ativo
b. Estático
c. Passivo
d. Dinâmico
41. O servidor recebe uma entrada de um *browser* através de _____.
a. Um atributo
b. Uma tag
c. Uma forma
d. Todas as alternativas anteriores
42. A saída de uma aplicação CGI é _____.
a. Texto
b. Gráfico
c. Dados binários
d. Todas as alternativas anteriores
43. Que tipo de documento da Web é transportado do servidor ao cliente na forma binária?
a. Estático
b. Dinâmico
c. Ativo
d. Todas as alternativas anteriores
44. Um *applet* é um pequeno programa escrito em _____.
a. C
b. C++
c. Shell script
d. Java
45. _____ é utilizada para habilitar o uso dos documentos ativos.
a. HTML
b. CGI
c. Java
d. Todas as alternativas acima
46. Java é _____.
a. Uma linguagem de programação
b. Um ambiente de alta performance
c. Uma biblioteca de classes
d. Todas as alternativas anteriores

47. Um *applet* é uma aplicação sob a forma de documento _____.
 a. Estático
 b. Ativo
 c. Passivo
 d. Dinâmico
48. Cotações de estoque são disponibilizados na Web. Este é provavelmente um documento _____.
 a. Ativo
 b. Estático
- c. Passivo
 d. Dinâmico
49. Dados atualizados sobre as coordenadas de satélite podem ser obtidos na WWW. Este é provavelmente um documento _____.
 a. Ativo
 b. Estático
 c. Passivo
 d. Dinâmico

Exercícios

50. Compare HTTP e FTP. Qual dos dois é mais simples? Explique sua resposta.
51. Compare o modo como os protocolos SMTP e HTTP transferem imagens. Qual deles é o mais eficiente? Por quê?
52. Que tipo de QoS é mais importante para o protocolo HTTP: atraso mínimo, throughput máximo ou confiabilidade? Qual é o menos importante? Explique suas respostas.
53. SMTP, FTP e HTTP são protocolos para transferência de mensagens entre dois pontos em uma rede. Compare e contraste-os.
54. O protocolo HTTP pode ser utilizado efetivamente para a transferência de áudio e vídeo armazenado? Explique sua resposta.
55. O protocolo HTTP pode ser utilizado efetivamente para a transferência de áudio e vídeo em tempo real? Explique sua resposta.
56. Um cliente HTTP pode monopolizar um servidor HTTP?
57. O protocolo HTTP realiza a comunicação entre cliente-servidor iterativa ou concorrentemente?
58. Apresente o resultado dos tags da linha da abaixo:
*Esta é
 uma linha escrita em
 HTML*
59. Apresente o resultado dos tags da linha abaixo:
*Esta é

 outra linha escrita em

 HTML*
60. Apresente o resultado dos tags das linhas abaixo:
*<H1> Documento </H1>
<H2> Este é um documento HTML
</H2>
<H1> Ele exibe o efeito dos tags
H </H1>*
61. Apresente o resultado dos tags das linhas abaixo:
*
 Sobrenome, Nome, Iniciais

 Endereço, Cidade
 Estado, CEP
*
62. Onde cada uma das figuras será exibida na tela?
*Olhe para a figura a seguir:
Então, responda-me o que você sente:

 Qual é a sensação? *
63. Apresente o resultado do segmento HTML abaixo:
*A editora deste livro é a

Editora Bookman *

Capítulo 28

Multimídia

Os avanços recentes da tecnologia estão modificando a forma como lidamos com os recursos de áudio e vídeo. Até bem pouco tempo limitávamos a ouvir um programa de rádio e/ou assistir programas de TV transmitidos por multidifusão (*broadcast*). Atualmente, queremos usar a Internet para usufruir dos serviços de áudio e vídeo que estão proliferando na rede e não só para transmitir mensagens de texto ou imagens estáticas. Neste último capítulo sobre a camada de aplicação, concentraremos nossa atenção nas aplicações que usam a Internet como canal de transmissão de aplicações de áudio e vídeo.

De acordo com a Figura 28.1, podemos dividir os serviços de áudio e vídeo em três grandes categorias: **streaming de áudio/vídeo armazenado**, **streaming de áudio/vídeo em tempo real** e **áudio/vídeo interativo**. O termo *streaming* significa que o usuário pode ouvir e/ou assistir um determinado arquivo após o *downloading* ter sido iniciado, isto é, no *streaming*, o conteúdo é executado sem que haja necessidade de trazê-lo por inteiro para o lado do cliente.

Nos serviços de *streaming* de áudio/vídeo armazenado (gravado) os arquivos sofrem um processo de compressão e são armazenados em um servidor. Um cliente faz o *download* dos arquivos através da Internet. Muitas vezes, este tipo de serviço é chamado **áudio/vídeo sob demanda**. Exemplos de arquivos de áudio armazenados: músicas, sinfonias, livros em *tape* e conferências. Exemplos de vídeos armazenados são filmes, programas de TV e *clips* de música.

O streaming de áudio/vídeo refere-se à demanda de compressão de arquivos de áudio/vídeo.

Nos serviços de *streaming* de áudio/vídeo em tempo real (ao vivo) um usuário escuta uma transmissão de áudio e/ou assiste uma transmissão de vídeo através da Internet em tempo real.



Figura 28.1 Áudio/vídeo na Internet.

Um excelente exemplo desse tipo de aplicação são as rádios na Internet. Algumas estações de rádio transmitem a programação delas somente na Internet; outras fazem transmissão tanto na Internet quanto via ondas de rádio. As transmissões de TV na Internet não são populares ainda, mas muitas pessoas estão apostando que, no futuro, a maioria das redes de TV que conhecemos hoje farão suas transmissões também na Internet.

O streaming de áudio/vídeo em tempo real refere-se às transmissões de programas de rádio e TV através da Internet.

Nos serviços de áudio/vídeo interativos um usuário usa a Internet para se comunicar interativamente com outro usuário. Bons exemplos dessa aplicação são o telefone via Internet ou as sessões de teleconferência.

O serviço de áudio/vídeo interativo refere-se ao uso da Internet para aplicações interativas de áudio/vídeo.

Neste capítulo analisaremos essas três aplicações. Entretanto, iniciaremos com duas questões relacionadas às aplicações de áudio/vídeo: *digitalização* e *compressão*.

28.1 DIGITALIZANDO ÁUDIO E VÍDEO

Antes dos sinais de áudio e/ou vídeo serem transmitidos através da Internet, precisamos digitalizá-los. Analisaremos as digitalizações de áudio e vídeo separadamente.

Áudio Digitalizado

Quando um sinal sonoro chega em um microfone é gerado um sinal elétrico que representa a amplitude do som como uma função do tempo. Este sinal é denominado *sinal de áudio analógico*. Vimos no Capítulo 5 que sinais analógicos, tal como o sinal de áudio, pode ser digitalizado produzindo um sinal digital. Aprendemos que, de acordo com o teorema de Nyquist, se a freqüência mais alta de um sinal composto é f , precisaremos amostrá-lo a 2 vezes f por segundo. Existem outros métodos para a digitalização de um sinal de áudio, mas o princípio básico é o mesmo. Limitaremos nossa discussão ao que foi tratado no Capítulo 5.

A voz é amostrada a 8000 amostras por segundo, utilizando 8-bits por amostra. Isto resulta em um sinal digital de 64 kbps. A música é amostrada a 44.100 amostras por segundo, utilizando 16-bits por amostra. Isto resulta em um sinal digital de 705,6 kbps (mono) ou 1.411 Mbps (estéreo).

Vídeo Digitalizado

Um sinal de vídeo consiste de uma seqüência de quadros. Se os quadros são mostrados na tela a uma velocidade satisfatória e na ordem cronológica dos eventos teremos a sensação de movimento. A razão fundamental é que nossos olhos não têm percepção sensorial suficiente para distinguir quadros individuais quando a velocidade de transição de quadros é suficientemente alta. Não existe um padrão para a quantidade de quadros por segundo exibidos em um tela. Nos Estados Unidos é comum o padrão de 25 quadros por segundo. Entretanto, para evitar que o brilho de uma imagem gráfica alterne com freqüência (processo denominado *flickering*), devido à baixa taxa de restauração da imagem ou à corrupção do sinal, um quadro precisa sofrer um processo de restauração (*refreshing*). A indústria de TV definiu a taxa de restauração de quadros em duas vezes. Isto significa que, nos Estados Unidos, 50 quadros precisam ser transmitidos ou, se houver memória no lado transmissor, 25 quadros com cada quadro restaurado a partir da memória.

Os quadros são divididos em pequenos pontos, denominados ***pixels***. Para uma imagem em preto e branco, 8 bits por pixel representam um dos 256 níveis diferentes de cinza. Para uma imagem colorida, cada pixel possui 24 bits, sendo que cada uma das cores primárias do padrão RGB (Red-Green-Blue) possui 8 bits.

Podemos determinar a quantidade de *bits* por segundo para uma resolução específica de um monitor de vídeo. Tomemos a resolução 1024×768 pixels. Para a transmissão de um quadro colorido nessa resolução precisamos:

$$2 \times 25 \times 1024 \times 768 \times 24 = 944 \text{ Mbps}$$

Nesta taxa de transmissão de dados precisamos de uma tecnologia que permita a transmissão de dados numa taxa muitíssimo alta, tal como a SONET. Para transmitir vídeo usando tecnologias de transmissão menos eficientes precisamos realizar a compressão de vídeo.

Vídeos precisam passar pelo processo de compressão para serem transmitidos pela Internet.

28.2 COMPRESSÃO DE ÁUDIO E VÍDEO

Para transmitir áudio e/ou vídeo pela Internet é necessário que o conjunto de dados digitalizados sofram **compressão**. Nesta seção discutiremos primeiramente a compressão de áudio e, em seguida, a compressão de vídeo.

Compressão de Áudio

A técnica de compressão de áudio pode ser utilizada na voz ou na música. No caso da voz, precisamos realizar a compressão do sinal digitalizado a 64 kHz. Para a música, a compressão do sinal deve acontecer a 1,411 MHz. Duas técnicas são usadas na compressão de áudio: codificação preditiva e codificação perceptiva.

Codificação Preditiva

Na **codificação preditiva** são codificadas as diferenças entre as amostras em vez de serem codificados todos os valores amostrados. Este tipo de compressão é usado normalmente para codificar voz. Muitos padrões foram definidos com essa finalidade; por exemplo, o GSM (13 kbps), G.729 (8 kbps) e G.723.3 (6,4 ou 5,3 kbps). Discussões detalhadas desses padrões estão fora do escopo deste livro.

Codificação Perceptiva: MP3

A técnica de compressão mais comum usada para criar CDs de áudio de qualidade é baseada na técnica de **codificação perceptiva**. Como mencionamos antes, este tipo de áudio precisa de pelo menos 1,411 Mbps para transmissão via Internet e, por isso, são necessários esquemas de compressão para transmiti-lo. O padrão **MP3** (MPEG-1 Layer 3), uma parte do padrão MPEG (discutida na seção sobre compressão de vídeo), usa esta técnica.

A codificação perceptiva usa a ciência da psicoacústica (ciência que estuda como as pessoas percebem o som) para estabelecer o método de codificação. A ideia baseia-se em algumas imperfeições do nosso sistema auditivo: alguns tipos de sons podem mascarar outros. Esse processo de mascaramento pode acontecer tanto no domínio da freqüência quanto no domínio do tempo. O **mascaramento em freqüência** acontece quando um som muito alto, em uma certa faixa de freqüências, mascara parcial ou totalmente um som mais baixo, noutra faixa de freqüências. Por exemplo, em um ambiente onde está sendo tocada música em alto volume temos bastante dificuldade em escutar o que as outras pessoas tentam nos comunicar. No **mascaramento temporal** um som alto pode anestesiá-los nossos ouvidos por um curto intervalo de tempo, até mesmo após o som ter cessado.

O MP3 usa estes dois fenômenos, mascaramento em freqüência e temporal, para comprimir sinais de áudio. A técnica de codificação analisa e divide o espectro de freqüência em muitos grupos. Os *bits* zero são associados às faixas de freqüência que foram totalmente mascaradas. Além disso, uma pequena quantidade de *bits* é associada às faixas de freqüências que foram parcial-

mente mascaradas e uma grande quantidade de *bits* é associada às faixas de freqüência não mascaradas.

O padrão MP3 produz três taxas de dados: 96 kbps, 128 kbps e 160 kbps. A taxa adotada baseia-se na faixa de freqüências do sinal analógico original.

Compressão de Vídeo

Conforme discutimos antes, o vídeo é composto de múltiplos quadros. Cada quadro é uma imagem. Podemos efetuar a compressão de vídeo comprimindo primeiramente as imagens. Na Internet prevalecem dois padrões de compressão de vídeo. O **Joint Photographic Experts Group (JPEG)** é usado para comprimir imagens. O **Moving Picture Experts Group (MPEG)** é usado para comprimir vídeo. Discutiremos rapidamente o padrão JPEG e, em seguida, o padrão MPEG.

Compressão de Imagem: JPEG

Como discutimos antes, se a imagem não é colorida, cada *pixel* pode ser representado por 8 *bits* (256 níveis de cinza). Se a imagem é colorida, cada *pixel* é representado em um padrão RGB de 24 *bits* (3×8 *bits*), onde cada *byte* representa um nível de vermelho, azul ou verde. Para simplificar a discussão, concentraremos nossa atenção na compressão de uma figura em escalas de cinza.

No padrão JPEG, uma figura em escala de cinza é dividida em blocos de 8×8 *pixels* (veja a Figura 28.2).

O propósito da divisão da figura em blocos é diminuir a quantidade de operações matemáticas envolvidas na transformação da figura porque, como veremos adiante, a complexidade matemática para cada figura varia com o quadrado do número de unidades.

Toda a idéia por trás do padrão JPEG é transformar a figura em um vetor que revela as redundâncias do processo. As redundâncias (repetições) podem ser removidas através de um dos métodos de compressão de texto. Uma versão simplificada do processo é mostrada na Figura 28.3.

Transformada Discreta Cosseno (DCT – Discrete Cosine Transform). Nesta etapa, cada bloco de 64 *pixels* sofre uma transformação denominada **transformada discreta de cossenos (DCT)**. A transformada DCT age nos 64 valores de cada bloco de tal forma que o relacionamento mútuo entre *pixels* é mantido, mas as redundâncias são reveladas. Não mostraremos a fórmula aqui, mas apresentaremos o resultado da transformação para três casos.

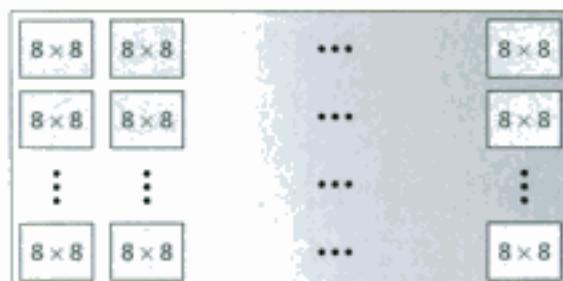


Figura 28.2 JPEG: escalas de cinza.

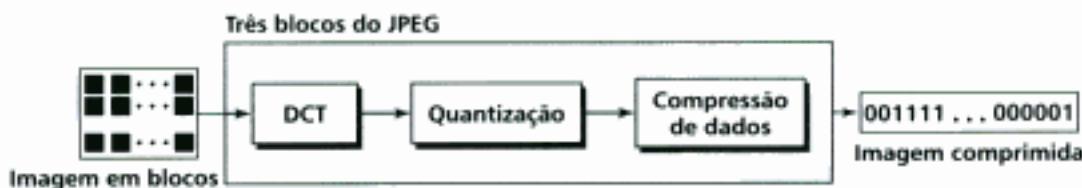


Figura 28.3 Processos JPEG.

Hidden page

Olhando as Figuras 28.4, 28.5 e 28.6 podemos afirmar o seguinte:

- A transformação DCT gera uma matriz T a partir da matriz P .
 - O valor dc é o valor médio dos valores dos *pixels* (multiplicado por uma constante).
 - Os valores ac representam as variações.
 - Todas as redundâncias nos *pixels* da vizinhança são transformadas em 0s.

Quantização Após a geração da matriz T , os valores são quantizados para reduzir a quantidade de bits necessária à codificação. Antes do processo de **quantização**, removemos a parte fracionária de cada valor e mantemos a parte inteira. Para fazer isso, dividimos o número por uma constante e, então, eliminamos a parte fracionária. Esse processo reduz ainda mais a quantidade de bits de codificação. Na maioria das implementações, uma matriz de quantização (8×8) define como quantizar cada valor na entrada da matriz T . Além disso, a escolha do divisor depende da posição do valor na matriz T . Isto otimiza o número de bits e o número de zeros para cada aplicação particular. Observe que a única fase no processo que não é reversível é a fase de quantização. Nessa etapa alguma parte da informação torna-se irrecuperável. Como resultado desse fato, às vezes, o padrão JPEG é denominado padrão de *compressão com perda*.

Compressão Após a quantização, os valores são lidos na matriz e os zeros redundantes são removidos. Entretanto, para agrupar os zeros juntos, a matriz é lida em zig-zague, diagonalmente, em vez de ser lida linha por linha ou coluna por coluna. A razão disso é que, se a figura não possui variações muito pequenas, a base da matriz T será toda composta de zeros. A Figura 28.7 ilustra o processo.

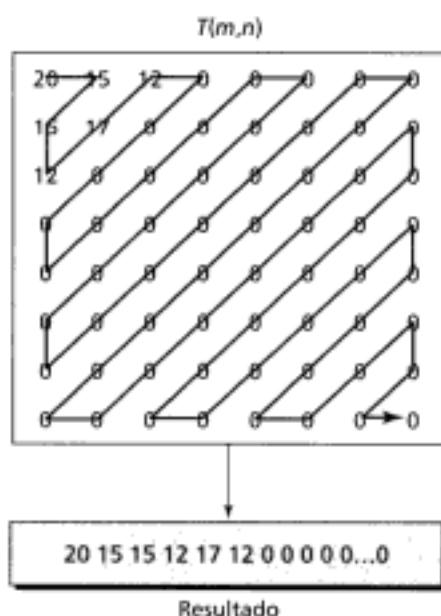


Figura 28.7 Leitura da tabela.

Compressão de Vídeo: MPEG

O método **Moving Picture Experts Group (MPEG)** é usado na compressão de vídeo. A idéia de movimento emerge da seqüência rápida de um conjunto de quadros devidamente ordenados, onde cada quadro é uma imagem estática. Em outras palavras, um quadro é uma combinação espacial de *pixels* e um vídeo é uma combinação temporal de quadros ordenados consecutivamente. Então,

Hidden page

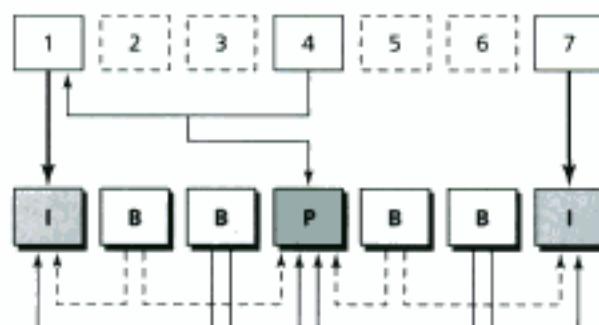


Figura 28.9 Construção do quadro MPEG.

28.3 STREAMING DE ÁUDIO E VÍDEO ARMAZENADO

Após a discussão dos processos de digitalização e compressão de áudio/vídeo, voltamos nossa atenção para aplicações específicas. A primeira aplicação trata de *streaming* de áudio/vídeo armazenado (gravado). O *downloading* desses tipos de arquivos de um servidor de Web na Internet pode ser diferente de outros tipos de arquivos. Para que você possa entender o conceito, vamos analisar quatro abordagens, cada qual com um grau de complexidade diferente.

Primeira Abordagem: Usando um Servidor de Web

O *download* de um arquivo de áudio/vídeo comprimido pode ser feito como um arquivo texto. O cliente (*browser*) pode usar os serviços do protocolo HTTP e enviar uma mensagem GET para iniciar o *download* do arquivo. Então, o servidor de Web responde enviando o arquivo comprimido para o *browser*. Nesse caso, o *browser* geralmente faz uso de um *helper* ou *plug-in*, caso não disponha desses recursos nativos, para exibir o arquivo. Um exemplo de *plug-in* é o **media player**. A Figura 28.10 ilustra a abordagem.

Esta abordagem é muito simples e não envolve o conceito de *streaming*. Contudo, ela tem uma desvantagem. Um arquivo de áudio/vídeo geralmente fica muito grande (em tamanho) após a compressão. Um arquivo de áudio pode facilmente superar algumas dezenas de *megabits* enquanto que um arquivo de vídeo alcança a casa das centenas de *megabits*. Nesta abordagem, um arquivo precisa ser baixado completamente antes de poder ser executado. Pensando nas taxas de transmissão de dados atuais, um usuário deve esperar alguns minutos antes de iniciar a exibição do arquivo de áudio/vídeo.



Figura 28.10 Usando um servidor de Web.

Hidden page

Real-Time Transport Control Protocol (RTCP)

O RTP permite somente um tipo de mensagem, uma que transporte dados da origem ao destino. Em muitos casos, há necessidade de outras mensagens em um sessão. Tais mensagens controlam o fluxo e a qualidade da informação transportada, permitindo ao receptor enviar informação de volta (*feedback*) à origem. O **Real-Time Transport Control Protocol (RTCP)** é um protocolo desenvolvido com esta finalidade. O RTCP possui cinco tipos de mensagens, conforme ilustra a Figura 28.19.

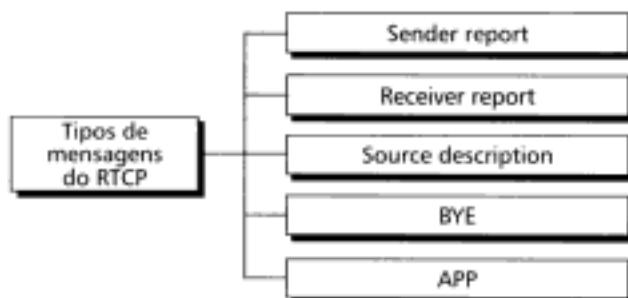


Figura 28.19 Tipos de mensagens RTCP.

Sender Report

A mensagem Sender Report (SR) é enviada periodicamente pelos transmissores ativos em uma sessão de multimídia para relatar as estatísticas de transmissão e recepção de todos os pacotes RTP enviados durante o intervalo. O SR inclui uma informação de *timestamp* absoluto na forma de um número, em segundos, transcorridos desde a meia-noite do dia 1 de janeiro de 1970. O *timestamp* absoluto permite ao receptor sincronizar mensagens RTP diferentes. Ele é particularmente importante quando tanto áudio quanto vídeo são transmitidos juntos (transmissões de áudio e vídeo usam informações separadas e relativas de *timestamps*).

Receiver Report

A mensagem Receiver Report (RR) destina-se aos participantes passivos, isto é, aqueles que não enviam pacotes RTP. A mensagem RR informa ao transmissor e aos outros receptores sobre a QoS da rede.

Source Description

As mensagens Source Description (SDES) são pacotes usados para controle da sessão que descrevem os parâmetros da fonte e contém o Canonical Name (CNAME), um identificador global usado para associar diferentes fluxos de mídia gerados pelo mesmo usuário. Esta informação pode ser o nome, endereço de *e-mail*, número de telefone e endereço do proprietário ou controlador da fonte.

BYE

Uma fonte envia uma mensagem BYE para desligar a sessão. Essa mensagem permite a uma fonte anunciar que ela está abandonando a sessão. Embora outras fontes possam detectar por si mesmas a saída de uma fonte particular, esta mensagem é um aviso direto. A mensagem BYE é muito útil ao *mixer*.

APP

A mensagem APP (*application-specific*) é um pacote para uma aplicação que deseja usar novas aplicações (não definidas no padrão). Em síntese, as mensagens APP adicionam funções específicas de uma aplicação ao pacote RTP.

Porta UDP

O protocolo RTCP, como o RTP, não usa um número de porta UDP conhecido. Logo, deve utilizar um número de porta temporário. A porta UDP escolhida deve ser o número imediatamente seguinte à porta UDP selecionada pelo RTP. Como o RTP utiliza um número de porta par, o RTCP deve utilizar um número de porta UDP ímpar.

O RTCP usa um número de porta UDP ímpar, que é o número de porta selecionado pelo RTP mais 1.

28.6 VOZ SOBRE IP (VOIP)

Vamos nos concentrar em uma aplicação específica de áudio/vídeo em tempo real: **voz sobre IP** ou **telefonia IP**. A idéia é utilizar a Internet como uma rede telefônica, com algumas funcionalidades adicionais. Em vez da comunicação acontecer em uma rede de comutação de circuitos, essa aplicação permite a comunicação entre duas partes em uma rede de comutação de pacotes (a Internet). Dois protocolos foram desenvolvidos para controlar este tipo de comunicação: SIP e H.323. Analisaremos resumidamente esses protocolos.

SIP

O **Session Initiation Protocol (SIP)** foi desenvolvido pelo IETF e logo foi publicado em RFC. O SIP é um protocolo de camada de aplicação que estabelece, gerencia e encerra uma sessão multimídia (uma chamada). Ele pode ser usado para criar sessões entre duas partes, multipartes ou *multicast*. O SIP foi criado para ser independente do suporte oferecido pela camada de transporte. Portanto, é suportado pelo UDP ou TCP.

Mensagens

O SIP é um protocolo baseado em texto, como o HTTP. O SIP, como o HTTP, usa mensagens. Seis mensagens estão definidas para esse protocolo, conforme ilustra a Figura 28.20.

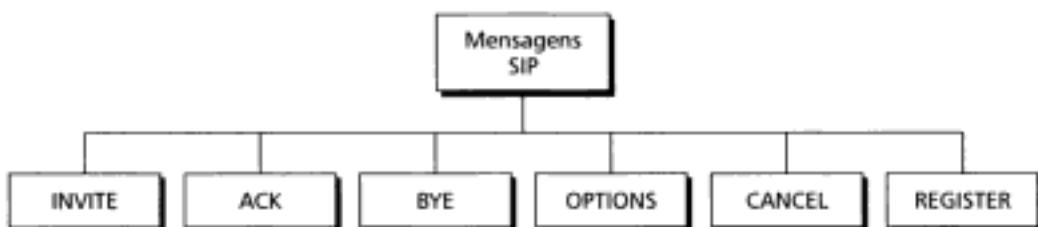


Figura 28.20 Mensagens SIP.

Cada mensagem possui um cabeçalho e um corpo. O cabeçalho consiste de muitas linhas que descrevem a estrutura da mensagem, como a capacidade do originador da chamada, tipo de meio e assim por diante. Daremos uma descrição breve dessas mensagens. Em seguida, mostraremos aplicações delas numa sessão de exemplos.

O originador da chamada inicializa a sessão através da mensagem INVITE. Após receber as respostas das partes chamadas, o originador transmite uma mensagem ACK para confirmação. Quando houver necessidade, a mensagem BYE termina a sessão. A mensagem OPTIONS, por sua vez, consulta uma máquina para obter informação sobre a capacidade específica dela. Como sugere o nome, a mensagem CANCEL cancela um processo já inicializado e, finalmente, a mensagem REGISTER é enviada pelo cliente para informar sua localização ao servidor.

Endereços

Em uma ligação telefônica tradicional uma série de números identifica o transmissor e, outra série de números, identifica o receptor. O protocolo SIP é bastante flexível com relação a isso. No SIP, um endereço de *e-mail*, um número de telefone e outros tipos de endereços podem ser utilizados para identificar o transmissor e o receptor. Porém, os endereços precisam estar no formato SIP (conhecido também como esquema SIP). A Figura 28.21 ilustra alguns formatos SIP típicos.



Figura 28.21 Formatos SIP.

Sessão Simples

Uma sessão simples usando o protocolo SIP consiste de três módulos: estabelecimento, comunicação e terminação da conexão. A Figura 28.22 mostra uma sessão simplificada.

Estabelecimento da Sessão O estabelecimento de uma sessão no protocolo SIP requer o mecanismo de *handshake* triplo. O originador envia uma mensagem INVITE, usando o UDP ou TCP, para iniciar a comunicação. Se a parte chamada estiver disposta a iniciar a sessão, é enviada uma mensagem de resposta. Por sua vez, o originador da chamada envia um ACK, para confirmar a recepção da mensagem de resposta.

Comunicação Após a sessão ter sido estabelecida, o originador e a parte chamada podem se comunicar usando números de porta temporários.

Término da sessão A sessão pode ser terminada com o envio da mensagem BYE por cada uma das partes.

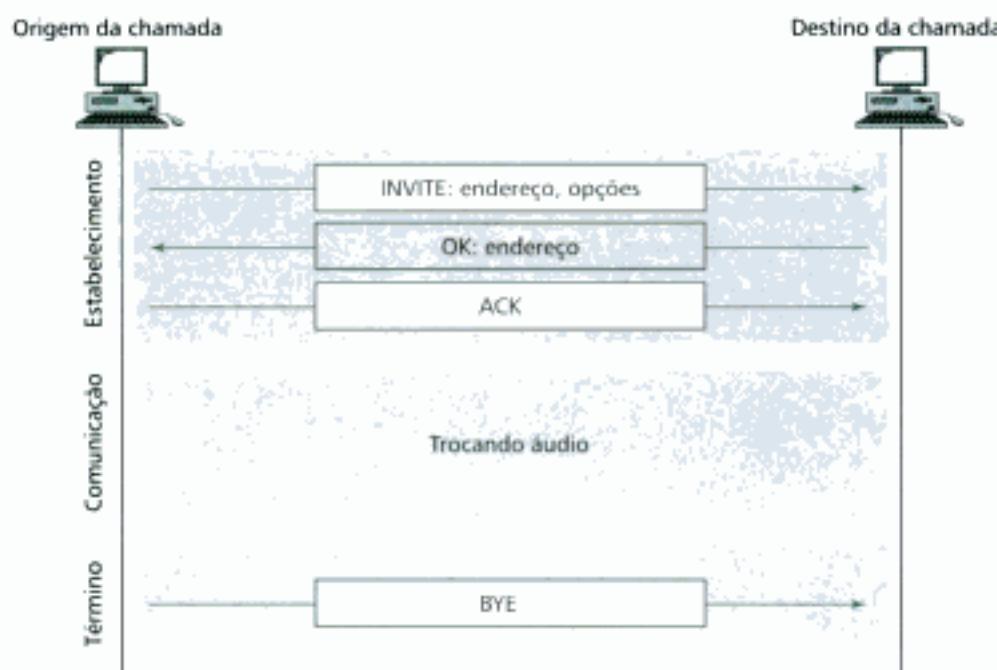


Figura 28.22 SIP: sessão simples.

Rastreando o Destino da Chamada

O SIP possui um mecanismo (similar ao DNS) que determina o endereço IP do terminal onde a parte chamada está situada. Para realizar este rastreamento, o SIP utiliza o conceito de registro ou inscrição. Além disso, o SIP define alguns servidores como registradores. Em qualquer instante de tempo um usuário é registrado em pelo menos um **servidor de registro**; este servidor conhece o endereço IP da parte chamada.

Quando o originador da chamada precisa comunicar-se com a parte chamada, ele pode utilizar o endereço de *e-mail*, em vez do endereço IP, na mensagem INVITE. Essa mensagem chega ao servidor de *proxy*. O servidor de *proxy* envia uma mensagem *lookup* (não faz parte do SIP) para algum servidor de registro que possui o endereço registrado da parte chamada. Quando o servidor de

proxy recebe a mensagem *reply* do servidor onde está localizado o registro, o servidor de *proxy* pega a mensagem INVITE do originador e insere nela o endereço IP da parte chamada, que acabou de ser descoberto. Então, essa mensagem é enviada à parte chamada. A Figura 28.23 apresenta as etapas do processo.

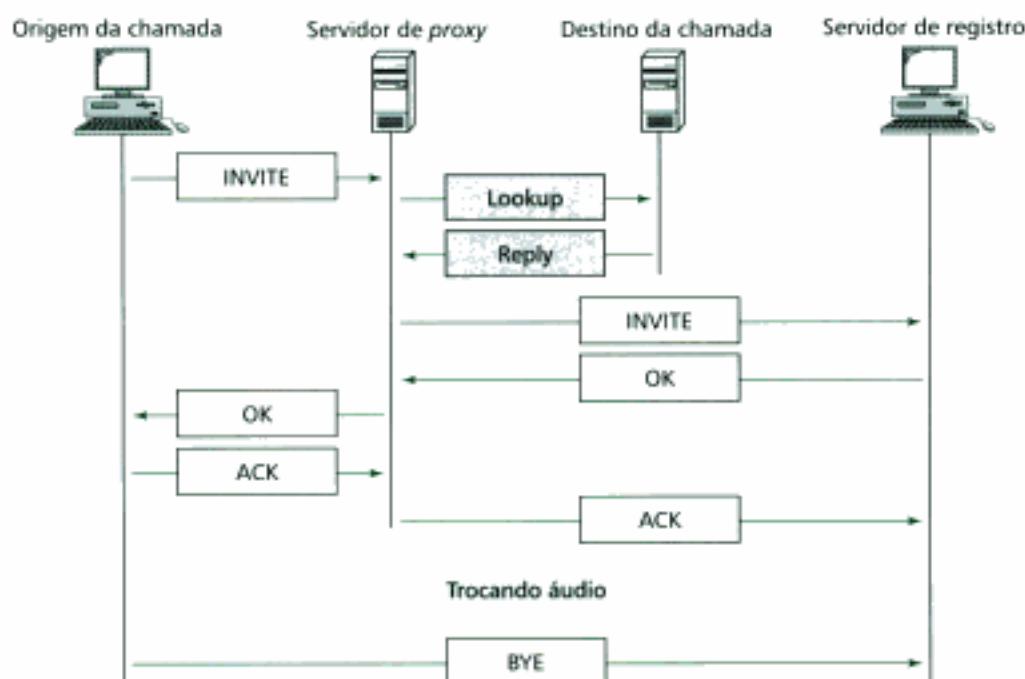


Figura 28.23 Rastreando o destino da chamada.

H.323

O protocolo **H.323** é um padrão projetado pelo ITU de maneira a permitir que os telefones da rede de telefonia pública conversem com computadores (chamados *terminais* no padrão H.323) conectados à Internet. A Figura 28.24 mostra a arquitetura geral do H.323.

Um **gateway** conecta a Internet à rede de telefonia pública. Em geral, um **gateway** é um dispositivo de cinco camadas que traduz uma mensagens de uma pilha de protocolos para outra. O conceito de **gateway** aqui não é exatamente a mesma coisa. Ele transforma uma mensagem padrão da rede telefônica em mensagem padrão Internet. O servidor **gatekeeper** localizado em uma rede local desempenha o papel de servidor de registro, conforme discutido no protocolo SIP.

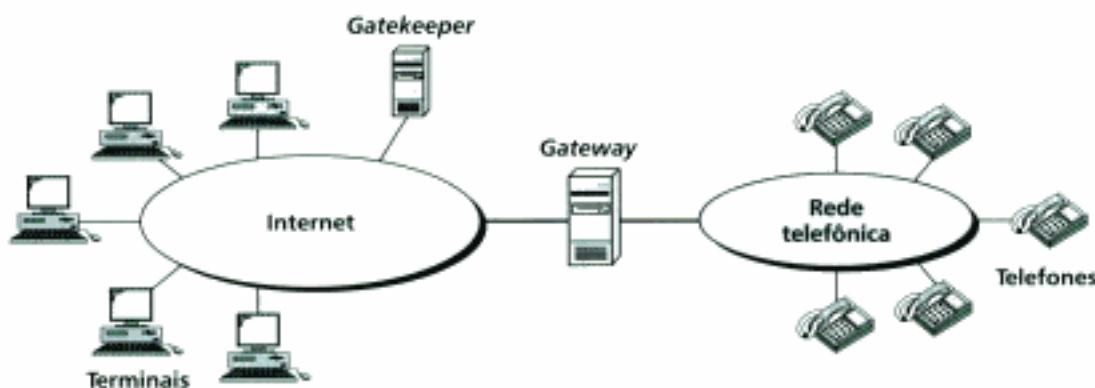


Figura 28.24 Arquitetura H.323.

Hidden page

Hidden page

Hidden page

Hidden page

Hidden page

PARTE VII

SEGURANÇA

Dedicamos a última parte do livro à segurança de rede, hoje um assunto fundamental dentro de um ambiente de rede. A segurança de rede é um assunto tão vasto e envolvente que, neste texto, faremos apenas uma introdução. A quantidade de livros tratando a questão da segurança na Internet tem crescido proporcionalmente à quantidade de *hackers*. Discutiremos somente os conceitos fundamentais relacionados à segurança.

Tópicos

Nesta parte do livro escolhemos muitos tópicos relacionados à segurança da informação na Internet. A Figura 1 mostra o relacionamento entre os tópicos de segurança.

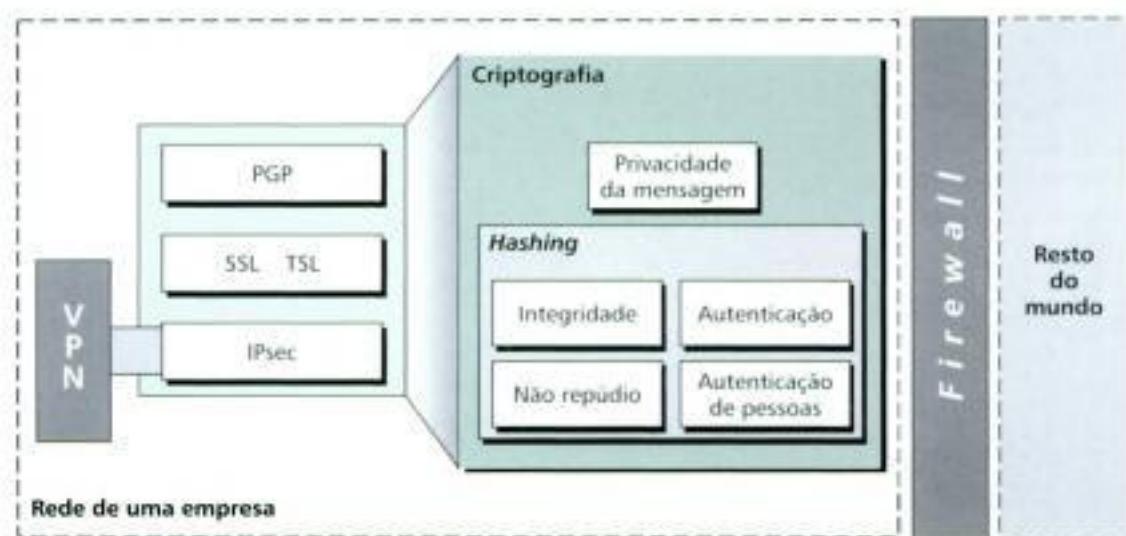


Figura 1 Tópicos de segurança.

Criptografia

A criptografia é o coração da segurança de rede. Se precisarmos estabelecer privacidade em uma rede, é de suma importância pensarmos como iremos criptografar a informação no transmissor e decodificá-la à forma original no receptor. Há muitos séculos, as pessoas acreditam que a criptografia requer o uso de uma chave secreta entre as duas partes. Recentemente, um novo método, a criptografia de chave pública, foi desenvolvida para utilizar duas chaves: uma pública e outra privada. Uma analogia para a criptografia de chave secreta ou privada é a chave que utilizamos para entrar em casa, a chave que abre uma porta é a mesma que a fecha novamente. Ao contrário, criptografia de chave pública requer os dois tipos de chave, uma de domínio comum e outra de domínio privado. Assim, olhando uma analogia, na criptografia de chave pública teríamos uma chave para abrir e outra para fechar a porta. Dedicaremos o Capítulo 29 aos esquemas de criptografia.

Aspectos da Segurança

Atualmente, segurança envolve muito mais do que simplesmente a privacidade de uma mensagem. Quando Alice envia uma mensagem para Bob ambos estão interessados em manter privacidade, mas eles também precisam se preocupar com outras questões. Ambos precisam estar certos de que a mensagem de Alice é realmente de Alice. Além disso, Bob precisa autenticar a mensagem. Bob também precisa estar certo de que a mensagem não foi corrompida durante a transmissão, ou seja, Bob precisa estar certo da integridade da mensagem. Se Bob representa um banco e Alice representa um cliente, Bob precisa ser capaz de provar mais tarde que foi mesmo Alice que enviou a mensagem, caso ela venha negar mais tarde. Isso é conhecido como não repúdio. Introduziremos estes aspectos de segurança no Capítulo 30.

Hashing

As funções de *hash* é um tópico diretamente ligado à criptografia. Hashing representa a criação de uma versão miniatura (minuta) de uma mensagem de forma a ser usada no lugar da mensagem original, para alguns aspectos de segurança. As funções de *hash* transformam um *input* de tamanho variável num *output* de tamanho (geralmente menor) fixo. Por exemplo, a minuta pode ter a integridade verificada. Se a minuta não tiver sido modificada durante uma transmissão, significa que a mensagem não foi corrompida. Discutiremos os conceitos por trás do hashing no Capítulo 30 quando tivermos tratando o problema da assinatura digital.

Autenticação de Usuários

Além da autenticação de mensagens, às vezes precisamos autenticar usuários (pessoas) ou processos para que eles possam ter acesso aos recursos de uma empresa ou organização. Tem havido muito desenvolvimento nessa área. Estudaremos algumas abordagens e daremos alguns critérios no Capítulo 30.

Gerenciamento de Chaves

Embora a criptografia se proponha a resolver alguns problemas relativos à segurança, ela cria outros numa rede tão grande quanto a Internet. Um desses problemas é o gerenciamento de chaves. Podemos trocar chaves secretas (privadas) com outros usuários que estejam a milhares de quilômetros? Podemos ter certeza de que a chave pública de Alice é realmente dela? O gerenciamento de chaves é estudado no Capítulo 30.

Arquitetura da Internet e Segurança

O modelo original da Internet provê um nível de segurança muito baixo. Por outro lado, o modelo OSI provê serviços de codificação/decodificação na camada de apresentação, uma camada que não existe na arquitetura da Internet. O modelo OSI nunca foi implementado.

Assim, a questão fundamental é: "se quisermos agregar segurança ao modelo da Internet, em qual nível ela deve ser agregada?" Os especialistas ainda não chegaram a uma conclusão definitiva, especialmente porque as falhas de segurança acontecem nas cinco camadas do modelo. Na camada física, um intruso pode derivar os sinais no meio de transmissão e ler ou alterar uma sequência de *bits*. Na camada de enlace, os *frames* podem ser capturados e lidos ou alterados. Isto é particularmente comum em uma LAN onde cada estação recebe uma cópia do *frame*. Na camada de rede, um datagrama IP pode ser removido, alterado ou inserido em uma rede. Na camada de transporte, um datagrama UDP ou um segmento TCP pode ser capturado ou alterado. Finalmente, na camada de aplicação, toda a mensagem pode ser alterada ou lida. Isso sugere que sejam respondidas as seguintes questões antes de aplicarmos ou agregarmos segurança à Internet.

1. Em que nível do modelo a segurança a ser implementada é mais efetiva?
2. Em que nível do modelo a segurança é mais fácil de ser implementada?
3. Se tivermos um bom nível de segurança em um nível, precisamos implementar segurança nos demais?

Segurança na Camada de Aplicação Vimos que a finalidade da Internet é fornecer comunicação entre processos finais. Esse argumento é utilizado como justificativa para implementar segurança somente na camada de aplicação. Se uma mensagem, indo de um processo para outro, é segura, os dois processos não se importam com o nível de segurança das outras camadas. Como sempre, há aqueles que vêm problemas nessa argumentação:

- a. Os programas aplicativos nesse nível estão muito bem estabelecidos e são independentes uns dos outros. Se quisermos implementar segurança nessa camada, precisamos fornecer segurança para cada protocolo de aplicação como, por exemplo, DNS, SMTP, FTP e HTTP. Isso significa que todas essas aplicações devem ser revistas. Embora isto envolva tempo e dinheiro, a idéia foi implementada no SMTP, como veremos no Capítulo 31.
- b. Nem toda comunicação na Internet é feita na camada de aplicação. Alguns serviços, tal como o RIP, usam os serviços da camada de transporte. Um intruso pode facilmente atacar o RIP e distorcer a mensagem atualizada para confundir e enganar os roteadores.
- c. Alguns tipos de protocolos usam os serviços do IP diretamente sem mesmo usar um protocolo da camada de transporte. Por exemplo, o OSPF entra nessa categoria.

Segurança na Camada de Transporte Muitas pessoas argumentam que a camada de transporte é a melhor camada para implementação da segurança. Podemos tomar medidas de segurança no UDP ou TCP. Toda aplicação usa o UDP ou TCP. Se esses dois estão em segurança, a aplicação também estará. Os oponentes a esta idéia argumentam que o UDP e o TCP são protocolos de transporte bem estabelecidos e mudá-los não seria uma tarefa trivial. Embora a segurança não tenha sido agregada diretamente ao UDP ou TCP, um protocolo de segurança denominado TLS foi desenvolvido para rodar na camada de transporte.

Segurança na Camada de Rede Muitos acreditam que uma ótima solução é implementar segurança na camada de rede, especificamente ao protocolo IP. Este argumento teve como resultado uma versão modificada do protocolo IP denominada IPsec (veja Capítulo 21).

Segurança na Camada de Enlace Embora existam argumentos plausíveis mostrando que a segurança pode ser facilmente implementada na camada de enlace visto que o domínio de ação é o *link*, muitos argumentos contrários afirmam que garantir segurança *node-to-node* não implica em garantir segurança *end-to-end*. Não discutiremos a segurança do *link* neste livro.

Firewalls

Em uma rede, embora possamos manter um excelente nível de confidencialidade das mensagens, preservar a integridade, autenticar o transmissor e assegurar o não repúdio de informação, estes aspectos de segurança sozinhos não impedem que uma pessoa mal-intencionada envie, deliberadamente, mensagens de modo a provocar danos em um sistema. Precisamos ainda de outra ferramenta. Precisamos de recursos para filtrar as mensagens de modo a permitir somente aquelas que nos interessam. O *firewall* é a tecnologia utilizada para esse fim. Estudaremos *firewalls* no Capítulo 31.

Virtual Private Networks (VPNs)

No passado, uma empresa ou organização que necessitasse de privacidade interna montava uma rede privada. Entretanto, se a empresa ou organização tivesse muitos braços espalhados mundo afora, conectar estas redes privadas usando WANs privadas envolvia um custo muito elevado. Hoje, empresas e organizações podem usar os serviços da Internet para conectar as redes privadas delas, onde a Internet desempenha o papel de uma rede WAN virtual privada. Estudaremos a tecnologia da rede virtual privada (Virtual Private Network – VPN) no Capítulo 31.

Organização dos Capítulos

A criptografia será discutida resumidamente no Capítulo 29. As questões relativas aos aspectos de segurança, autenticação de usuários e gerenciamento de chaves são discutidas no Capítulo 30. Finalmente, cobriremos segurança na Internet, *firewalls* e VPNs no Capítulo 31.

Hidden page

to cifrado. Um algoritmo de decifragem reverte o processo, transformando o texto cifrado em texto limpo. O transmissor usa um algoritmo de cifragem, enquanto que o receptor usa um algoritmo de decifragem.

Ao longo deste e dos Capítulos 30 e 31, discutiremos algoritmos de cifragem e decifragem. Iremos nos referir a eles como **cifras**. O termo *cifra* é utilizado também quando nos referimos aos diferentes tipos de algoritmos de criptografia.

Não é necessário que cada par transmissor-receptor tenha uma cifra única própria para comunicação segura. Ao invés disso, através do uso de cifras públicas com chaves secretas, uma cifra pode servir milhões de pares transmissor-receptor. Uma **chave** é um número (valor) sobre o que a cifra, enquanto algoritmo, deve operar. Para cifrar uma mensagem, precisamos de um algoritmo de cifragem, uma chave e um texto limpo. Desse conjunto origina-se o texto cifrado. Para decifrar uma mensagem, precisamos de um algoritmo de decifragem, uma chave e um texto cifrado. Este conjunto revela o texto limpo original. A Figura 29.2 ilustra a idéia.

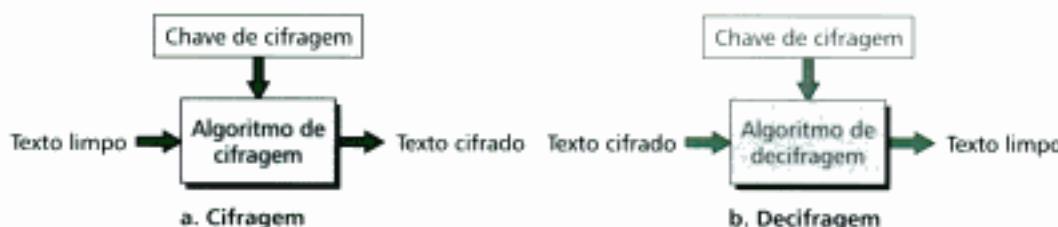


Figura 29.2 Cifragem e decifragem.

Os algoritmos de cifragem e decifragem são públicos, qualquer um pode acessá-los. As chaves são secretas, elas precisam ser protegidas.

Na criptografia, os algoritmos de cifragem/decifragem são públicos. As chaves são secretas.

É comum introduzirmos três personagens (nomes) na criptografia: Alice, Bob e Eve. Alice é a personagem que necessita transmitir dados com segurança. Bob é o personagem receptor dos dados. Eve é a personagem que, de algum modo, perturba a comunicação entre Alice e Bob, interceptando mensagens ou enviando mensagens dissimuladas próprias. Estes três personagens representam computadores ou processos que de fato enviam, recebem, interceptam ou modificam dados.

Todos os algoritmos de criptografia* desenvolvidos foram separados em dois grupos: algoritmo de criptografia com chave simétrica (às vezes chamada chave secreta ou chave privada) e os algoritmos de criptografia com chave pública (às vezes chamados chave assimétrica).

29.2 CRIPTOGRAFIA COM CHAVE SIMÉTRICA

Na **criptografia com chave simétrica**, tanto a Alice quanto o Bob usam a mesma chave. A Alice usa essa chave e o algoritmo de cifragem para cifrar a mensagem. O Bob usa a mesma chave e um algoritmo de decifragem correspondente para decifrar a mensagem (veja a Figura 29.3).

Na criptografia com chave simétrica, a mesma chave é utilizada pela Alice (cifragem) e pelo Bob (decifragem). A chave é compartilhada.

* N. de R. T.: Quando o autor se refere à criptografia, ele quer dizer criptografia clássica. Quando são tratados os algoritmos de criptografia quânticos são necessários alguns cuidados adicionais para incluir a superposição de estados descrita na teoria quântica.

Hidden page

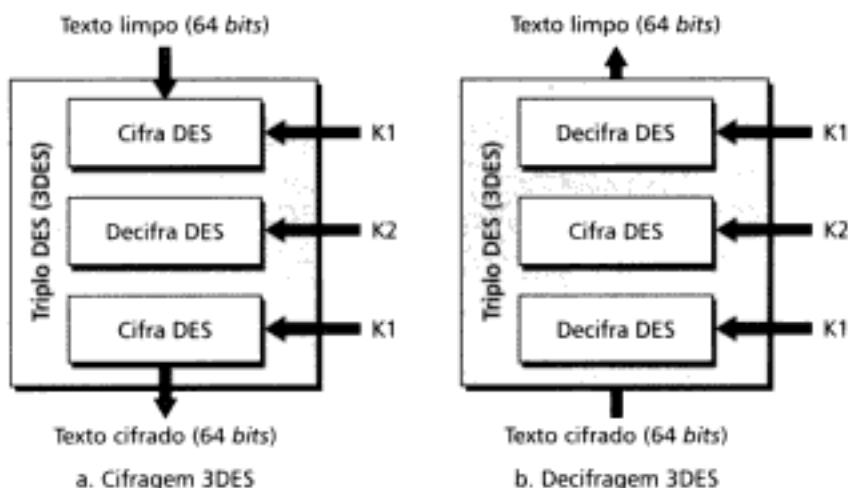


Figura 29.15 Triplo DES (3DES).

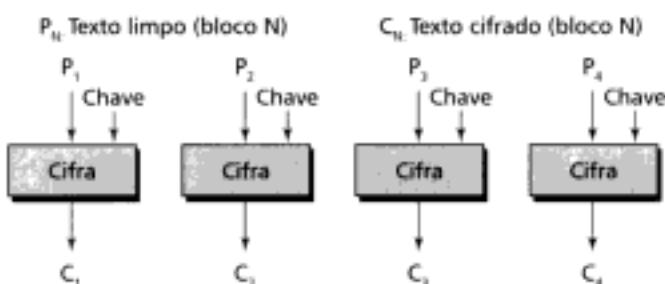


Figura 29.16 Modo ECB.

dependente dos outros blocos do código ECB. Embora a figura mostre somente quatro blocos, o ECB foi projetado para controlar muitos mais.

O problema com o modo ECB é que a cifragem de cada bloco de 8 bytes é independente dos demais; isto é, a cifragem de cada bloco não depende dos outros blocos no processador. Isto significa que Eve poderia trocar dois blocos e Bob não notaria esta mudança, se ambos blocos estiverem relacionados à mesma mensagem. Por exemplo, se Eve sabe que os blocos 4, 8, 12, 16, ..., representam as notas médias de um estudante, Eve poderia permutar o bloco 8 com o bloco 16 (substituir uma nota mais baixa por outra mais alta).

Cipher Block Chaining

No modo **Cipher Block Chaining (CBC)**, a cifragem ou decifragem de um bloco depende de todos os blocos anteriores, como mostra a Figura 29.17.

Por exemplo, para cifrar o segundo bloco de texto limpo (P_2) primeiramente aplicamos uma operação XOR de P_2 com o primeiro bloco de texto cifrado (C_1) e então o passamos através do processo de cifragem. Desse modo, C_2 depende de C_1 . Se Eve tentar trocar C_1 por C_3 , por exemplo, C_2 não será decifrado corretamente. A situação para o primeiro bloco é diferente porque não existe uma entrada C_0 . Em vez disso, é usado um número aleatório de 64 bits, chamado *vetor de inicialização (IV)*. O IV é enviado juntamente com os dados para que o receptor possa usá-lo na decifragem.

Cipher Feedback Block

O **Cipher Feedback Mode (CFM)** foi desenvolvido para aquelas situações onde precisamos enviar ou receber 1 byte de dados por vez, mas ainda queremos usar o DES ou triplo DES. Uma solução é fazer 1 byte C_n dependente de 1 byte P_n e outro byte aleatório, o qual depende dos 8 bytes anteriores, como mostra a Figura 29.18.

Hidden page

Hidden page

RSA

O método mais utilizado na criptografia com chave pública é denominado **método RSA** devido aos inventores Rivest, Shamir e Adleman. Nesse método, a chave privada é um par de números (N, d) assim como a chave pública (N, e). Observe que o número N é comum às chaves pública e privada.

Alice usa o seguinte algoritmo para cifrar a mensagem:

$$C = P^e \bmod N$$

Neste algoritmo, P é o texto limpo, que é representado como um número; C é o número que representa o texto cifrado. Os dois números e e N são os componentes da chave pública. O texto limpo é gerado através de $P = C^d \bmod N$. O termo mod indica que o resto da divisão é enviado como texto cifrado.

Bob usa o seguinte algoritmo para decifrar a mensagem:

$$P = C^d \bmod N$$

Neste algoritmo, as variáveis P e C são as mesmas descritas acima. Os dois números d e N são os componentes da chave privada. A Figura 29.21 mostra um exemplo.

Imagine que a chave privada seja o par $(119, 77)$ e que a chave pública seja o par $(119, 5)$. Imagine ainda que Alice precisa transmitir o caractere F. Este caractere pode ser representado como o número 6 (F é o sexto caractere do alfabeto). O algoritmo de cifragem calcula $C = 6^5 \bmod 119 = 41$. Este número é enviado ao receptor como o texto cifrado. O receptor usa o algoritmo de decifragem para determinar $P = 41^{77} \bmod 119 = 6$ (o número original). O número 6 é então representado como a letra F.

O estudante pode perguntar sobre a eficiência desse algoritmo. Se Eve conhece o algoritmo de decifragem e $N = 119$, a única informação que Eve desconhece é o valor de $d = 77$. Então, por que Eve não usa o método de tentativa e erro para determinar d ? Ela poderia sim. No exemplo citado anteriormente, facilmente Eve chegaria ao valor correto de d . Entretanto, a força do algoritmo RSA responde na utilização de números primos grandes para d e e . Na prática, os números são muito grandes (na faixa de 10 algarismos) que a possibilidade de quebra do algoritmo por tentativa e erro levaria muito tempo (meses, talvez anos) até mesmo para os computadores mais rápidos disponíveis atualmente.

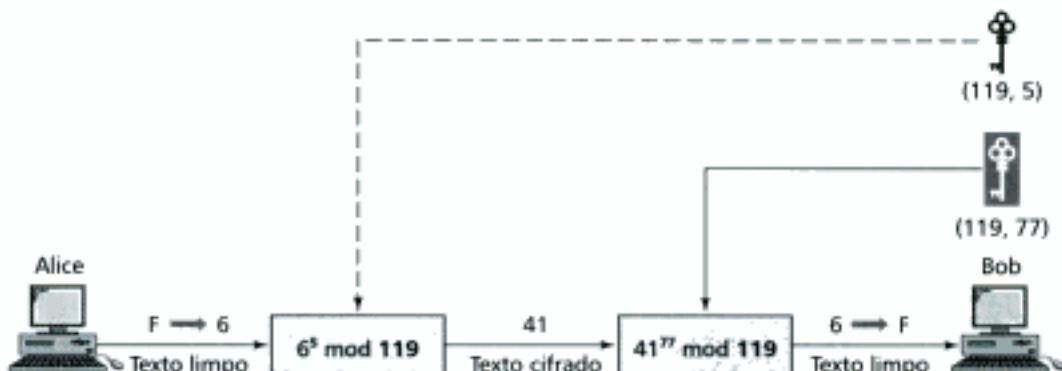


Figura 29.21 RSA.

Hidden page

Hidden page

- b. Modo CBC
 - c. Modo CFM
 - d. Todos os modos acima
23. Um vetor de inicialização é necessário no _____.
- a. Modo ECB
 - b. Modo CBC
 - c. CVF
 - d. Modo CSM
24. No _____ a cifragem de cada bloco de 8 bytes é independente das demais.
- a. Modo ECB
 - b. Modo CBC
 - c. CVF
 - d. Modo CSM
25. No método de criptografia com chave pública, que chave é conhecida publicamente?
- a. Somente a chave privada
 - b. Somente a chave pública
 - c. Ambas chaves
 - d. Nenhuma das alternativas acima
26. No método de criptografia com chave pública, somente Bob tem em seu poder a chave _____.
- a. Privada
 - b. Pública
 - c. Ambas chaves
 - d. Nenhuma das alternativas acima
27. O algoritmo RSA usa um método de criptografia com _____.
- a. Chave pública
 - b. Chave privada
 - c. Chave simétrica
 - d. Nenhuma das alternativas anteriores

Exercícios

28. Cifre a mensagem a seguir usando substituição monoalfabética com chave = 4.
ESTE É UM BOM EXEMPLO
29. Decifre a mensagem a seguir usando substituição monoalfabética com chave = 4.
IRGVCTXMSR MW JYR
30. Decifre a mensagem a seguir, usando substituição monoalfabética sem o conhecimento da chave (obs.: a mensagem está em Inglês).
KTIXEVZOUT OY ROQK KTIRUYOTM G YKIXKZ OT GT KTBKRUVK
31. Cifre a mensagem a seguir, usando substituição polialfabética. Use a posição de cada caractere como chave.
UM MAIS UM É DOIS, UM MAIS DOIS É TRÊS, UM MAIS TRÊS É QUATRO.
32. Use o algoritmo de cifragem abaixo para cifrar a mensagem "BOM DIA".
- a. Substitua cada caractere pelo seu valor no código ASCII.
 - b. Adicione um bit 0 à esquerda para escrever cada caractere em 8 bits.
 - c. Permute os 4 primeiros bits com os 4 últimos.
 - d. Substitua cada grupo de 4 bits pelo equivalente em hexadecimal.

- Qual é a chave deste método?
33. Use o algoritmo de cifragem a seguir para cifrar a mensagem "ABCDEF GH" (assuma que a mensagem é feita apenas de caracteres maiúsculos).
- a. Trate cada caractere como um número decimal, usando o código ASCII (entre 65 e 90).
 - b. Subtraia 65 de cada caractere codificado.
 - c. Converta cada número para binário com 5 bits.
34. Usando o algoritmo RSA, cifre e decifre a mensagem "BE" usando o par de chaves (3,15) e (5,15).
35. Dados dois números primos $p = 19$ e $q = 23$, tente determinar N , e e d .
36. Para verificar o nível de segurança do algoritmo RSA, determine d sabendo que $e = 17$ e $N = 187$.
37. No algoritmo RSA, usamos $C = P^e \bmod N$ para cifrar um número. Se e e N forem números grandes (cada um com centenas de algarismos), o cálculo é impossível e gera um erro de overflow (estouro) até mesmo em um supercomputador. Uma solução (não é a melhor) usando a teoria dos números envolve muita etapas, onde cada passo utiliza o resultado gerado no passo anterior:
- a. $C = 1$.

Hidden page

Segurança da Informação, Autenticação de Usuários e Gerenciamento de Chaves

Após o estudo do Capítulo 29, analisaremos algumas aplicações da criptografia: segurança da informação, autenticação de usuários e gerenciamento de chaves.

A segurança da informação envolve confidencialidade, integridade, autenticação e, finalmente, o não repúdio (a rejeição).

Autenticação do usuário significa verificar a identidade da pessoa ou do processo que deseja se comunicar com o sistema protegido. A autenticação dos usuários também é necessária no gerenciamento de chaves.

Finalmente, precisamos do gerenciamento de chaves: a distribuição das chaves simétricas e a certificação das chaves públicas. A Seção 30.4 explica os métodos utilizados no gerenciamento de chaves.

30.1 SEGURANÇA DA INFORMAÇÃO

Vamos primeiramente discutir as medidas de segurança aplicadas à informação. Podemos dizer que a segurança deve garantir quatro serviços: privacidade (confidencialidade), autenticação, integridade e o não repúdio (a rejeição). A Figura 30.1 ilustra o arcabouço da segurança da informação.



Figura 30.1 Segurança da informação.

Privacidade

Privacidade significa que tanto Alice quanto Bob contam com confidencialidade. A mensagem transmitida deve fazer sentido, isto é, chegar limpa, somente para o Bob. Para terceiros quaisquer, a mensagem deve ser ininteligível.

A forma como podemos obter privacidade é a mesma há séculos: a mensagem deve ser criptografada. Uma boa técnica de privacidade assegura, com algum nível de segurança, que um ataque potencial de Eve, que estava bisbilhotando a comunicação, possa comprometer o conteúdo da mensagem.

Privacidade Usando Criptografia com Chave Simétrica

A privacidade pode ser obtida usando a cifragem/decifragem com chave simétrica, conforme ilustra a Figura 30.2. De acordo com a discussão do Capítulo 29, na criptografia com chave simétrica a chave é compartilhada entre Alice e Bob.

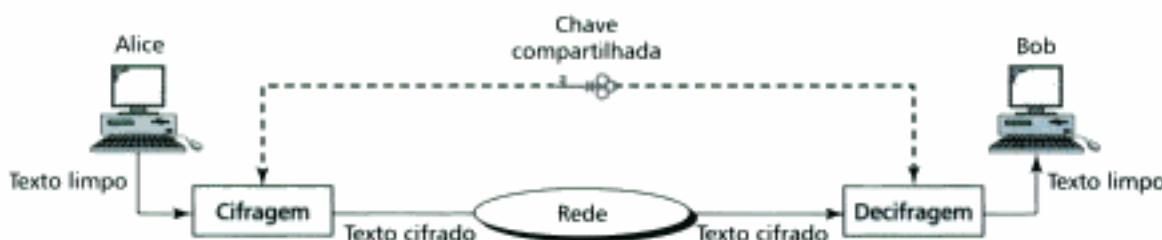


Figura 30.2 Privacidade usando criptografia com chave simétrica.

O uso da criptografia com chave simétrica é muito comum na tentativa de adicionar privacidade à comunicação. Mais tarde, neste capítulo, veremos como gerenciar a distribuição de chaves simétricas.

Privacidade Usando Criptografia com Chave Pública

Também podemos obter privacidade usando a cifragem com chave pública. Este tipo de privacidade envolve o uso de duas chaves: a chave privada e a chave pública. A chave pública é, como diz o nome, de domínio público. Isto aparece ilustrado na Figura 30.3.

O problema principal da cifragem usando a chave pública é que o dono da mensagem deve ter a identidade verificada (certificada). Veremos como tratar esse problema em breve.



Figura 30.3 Privacidade usando criptografia com chave pública.

Autenticação da Mensagem

Autenticação da mensagem significa que Bob deve ter certeza da identidade de Alice e que Eve não tenha enviado uma mensagem tentando se passar por Alice. Veremos como a assinatura digital pode fornecer autenticação de mensagem.

Integridade

Integridade de dados significa que os dados devem chegar ao Bob exatamente como eles foram enviados por Alice. Não podem ocorrer mudanças durante a transmissão, seja acidental ou maliciosa. Por exemplo, o volume das movimentações financeiras através da Internet é algo vertiginoso. Por isso, integridade é crucial. Seria um desastre para Alice se ela estivesse transferindo R\$100,00 para a conta de Bob e tivesse R\$10.000,00 ou R\$100.000,00 debitados de sua conta (é claro, supondo que Alice disponha dessas quantias). A integridade da informação deve ser preservada em uma comunicação segura. Veremos como a assinatura digital pode agregar integridade à informação.

O Problema do Não Repúdio

O **não repúdio** significa que Bob deve ser capaz de provar que a informação recebida de Alice veio realmente dela. Além disso, Alice não deve ser capaz de negar (repudiar) o envio da informação que ela de fato enviou. O ônus da prova recai sobre Bob. Por exemplo, quando um usuário envia uma mensagem para transferir dinheiro de uma conta para outra, o banco deve provar que o usuário realmente requisitou esta transação financeira. Veremos que a assinatura digital também pode resolver o problema do não repúdio (rejeição).

30.2 ASSINATURA DIGITAL

Vimos que segurança está relacionada a quatro características da informação: privacidade, autenticação, integridade e o não repúdio. Fizemos uma discussão genérica da privacidade. Para discutirmos as outras três características, precisamos olhar mais de perto o conceito da **assinatura digital**.

A idéia é similar a assinatura em um documento de identificação. Quando transmitimos um documento eletronicamente, devemos assiná-lo. Temos duas escolhas para isso: podemos assinar todo o documento ou assinar uma versão resumida ou sintetizada do documento (*digest message*).

Assinando um Documento

A cifragem com chave pública pode ser utilizada para assinar um documento. Entretanto, os papéis das chaves pública e privada são diferentes nesse caso. Alice usa a chave privada dela para cifrar (assinar) a mensagem, assim como uma pessoa usa assinatura dela para validar um documento. Por outro lado, Bob usa a chave pública de Alice para decifrar a mensagem, assim como uma pessoa pode verificar assinatura de outra pessoa, por exemplo, olhando o registro de firma em cartório.

Na assinatura digital, a chave privada é usada para cifragem e a chave pública é usada para decifragem da mensagem. Isto é possível porque os algoritmos de cifragem/decifragem atuais, tal como o RSA, são fórmulas matemáticas e suas estruturas são similares. A Figura 30.4 mostra como a idéia é implementada na prática.

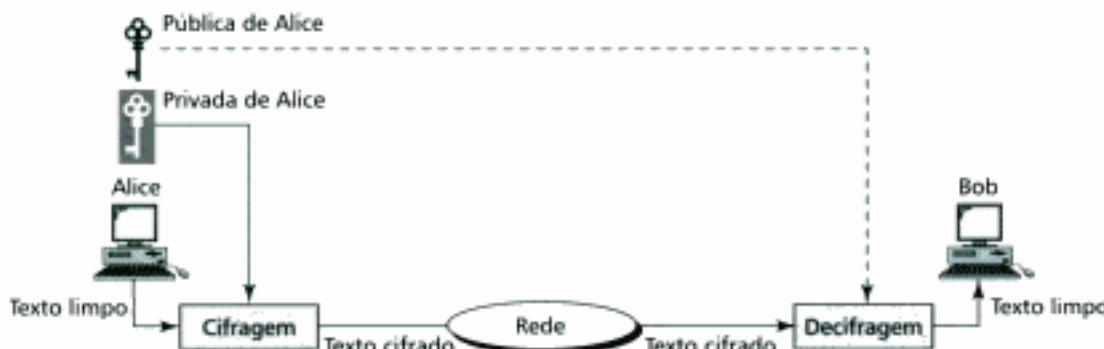


Figura 30.4 Assinando um documento.

A assinatura digital pode agregar integridade, autenticação e o não repúdio.

Integridade A integridade de uma mensagem é preservada porque se Eve interceptar e modificá-la, parcial ou totalmente, a mensagem decifrada seria inteligível para ela.

Autenticação Podemos usar o seguinte raciocínio para mostrar como uma mensagem pode ser autenticada. Se Eve envia uma mensagem se fazendo passar por Alice, mas não usa a chave privada de Alice para cifragem, porque ela desconhece, Bob usará a chave pública de Alice para decifrar a mensagem e obterá um resultado inteligível, porque foi Eve quem enviou a mensagem e não Alice. Assim, a cifragem com a chave privada de Eve e a decifragem com a chave pública de Alice é lixo para o Bob.

Não Repúdio A assinatura digital também fornece recursos à não rejeição de mensagens. Bob recebe a mensagem de Alice e a salva. Depois, se Alice negar o envio da mensagem, Bob é capaz de mostrar que a cifragem/decifragem da mensagem salva com as chaves privada e pública de Alice gera uma duplicata da mensagem salva. Como somente Alice conhece a chave privada dela, ela não pode repudiar o envio da mensagem.

A assinatura digital sozinha não agrega privacidade. Se o nível de privacidade necessário for muito rigoroso, outro nível de cifragem/decifragem deve ser aplicado.

Assinando a Síntese de um Documento

Dissemos anteriormente que a criptografia com chave pública é bastante eficiente com mensagens pequenas. O uso da chave pública para assinar toda a mensagem torna o processo pouco eficiente se o tamanho da mensagem for muito grande. A solução é permitir que Alice assine uma minuta do documento em vez de assinar o documento inteiro. Alice cria uma versão resumida do documento (*digest*) e o assina. Então, Bob verifica a assinatura na versão resumida ou sintetizada.

Para criar uma síntese da mensagem (*digest*) usamos uma **função de hash** (função de síntese). A finalidade da função de *hash* é criar uma síntese da mensagem de tamanho fixo a partir da mensagem original de tamanho variável, conforme mostra a Figura 30.5.

As duas funções de *hash* mais importantes são denominadas MD5 (Message Digest 5) e SHA-1 (Secure Hash Algorithm 1). A primeira produz uma versão sintetizada de 120 bits de tamanho. A segunda produz uma versão de 160 bits.



Figura 30.5 Assinando a síntese (*digest*).

Hidden page

Hidden page

Segunda Abordagem

Para evitar um ataque *replay*, adicionamos informação ao procedimento para ajudar Bob a distinguir entre uma autenticação nova e uma autenticação repetida (clonada). Isto é feito usando o ***nonce***. Um *nonce* é um número aleatório grande que é usado somente uma vez. Nessa abordagem, Bob usa um *nonce* para desafiar Alice, de modo a ter certeza de que Alice é autêntica e que Eve não está tentando personificá-la. A Figura 30.9 ilustra o procedimento.

A autenticação acontece em três etapas. Primeiro, Alice envia, em um texto limpo, a identidade dela ao Bob. Bob desafia Alice a confirmar a identidade dela, enviando um *nonce* R_B em texto limpo. Alice responde ao desafio enviando de volta o *nonce*, cifrado através da chave simétrica dela. Eve não pode responder ao desafio visto que R_B é válido uma única vez.



Figura 30.9 Usando um *nonce*.

Autenticação Bidirecional

A segunda abordagem de uma solicitação e uma resposta de autenticidade de Alice a Bob é a autenticação bidirecional. É possível termos **autenticação bidirecional**? A Figura 30.10 ilustra um método.

Na primeira etapa, Alice envia a identificação e o *nonce* dela desafiando Bob. Na segunda etapa, Bob responde ao desafio de Alice enviando o *nonce* dele (cifrado) e desafiando Alice. Na terceira etapa, Alice responde ao desafio de Bob. Esta autenticação é totalmente segura? Ela é segura considerando que Alice e Bob usam conjuntos de *nonces* diferentes para sessões diferentes e que não existam autenticações múltiplas ao mesmo tempo. De outro modo, este procedimento pode ser alvo de um ataque denominado **reflexão**. Deixamos isso como um exercício.

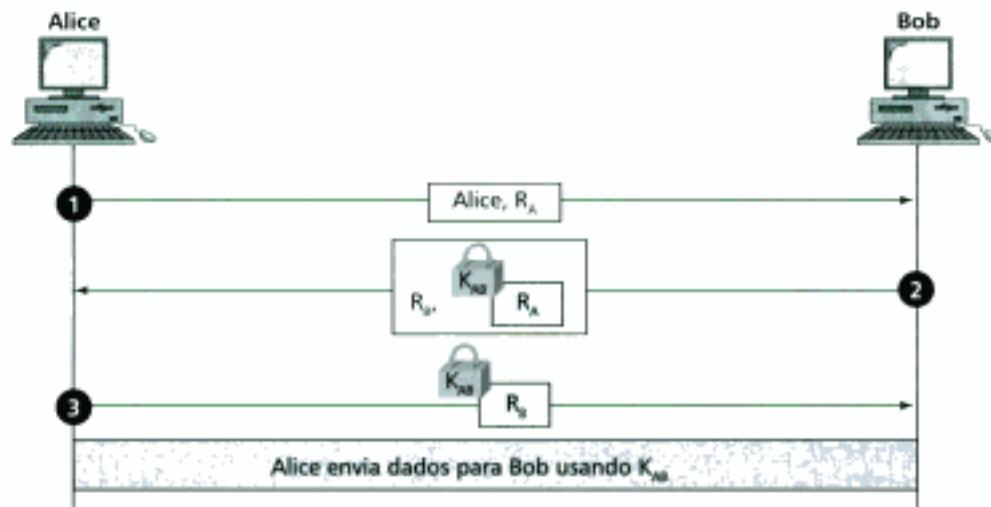


Figura 30.10 Autenticação bidirecional.

Hidden page

so em ter que armazená-las para uso futuro. As partes não precisam entrar em nenhum tipo de acordo sobre a chave. Vamos estudar como o protocolo trabalha quando Alice e Bob precisam de uma chave simétrica para se comunicarem.

Pré-Requisito Antes do estabelecimento de uma chave simétrica, Bob e Alice precisam escolher dois números. Sejam N e G estes números. O primeiro número (N) é um número primo grande com a restrição de que $(N - 1)/2$ deve ser um número primo também. O segundo número (G) deve ser outro número primo, mas ele não tem restrições. Estes números não precisam ser confidenciais. Eles podem ser passados através da Internet ou estampados em um *outdoor* de propaganda, pois eles são públicos. Dois números quaisquer, escolhidos apropriadamente, podem servir todas as aplicações do mundo inteiro. Assim não há nenhum tipo de segredo em relação a estes números e, portanto, tanto Alice quanto Bob têm conhecimento deles.

Procedimento A Figura 30.11 mostra o procedimento.

As etapas são as seguintes:

Etapa 1 Alice escolhe um número aleatório grande (x) e, a partir dele, determina $R1 = G^x \text{ mod } N$.

Etapa 2 Alice transmite o número $R1$ para o Bob. Note que Alice não envia o valor x . Ela envia apenas $R1$.

Etapa 3 Bob escolhe outro número aleatório grande (y) e, a partir dele, determina $R2 = G^y \text{ mod } N$.

Etapa 4 Bob transmite $R2$ para Alice. Novamente, note que Bob não envia o valor y para Alice; ele envia apenas $R2$.

Etapa 5 Alice determina $K = (R2)^x \text{ mod } N$. Bob também determina o mesmo K , através de $K = (R1)^y \text{ mod } N$. O número K é a chave simétrica desejada.

Você deve ter ficado demasiadamente estupefato porque o valor K é o mesmo nos dois cálculos. A resposta está na igualdade demonstrada na teoria dos números.

$$(G^x \text{ mod } N)^y \text{ mod } N = (G^y \text{ mod } N)^x \text{ mod } N = G^{xy} \text{ mod } N$$

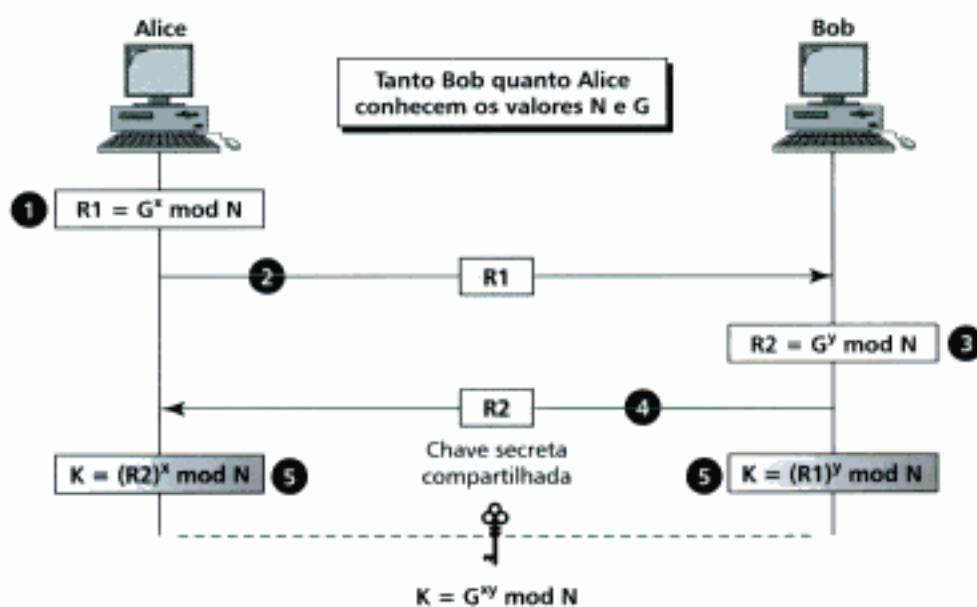


Figura 30.11 Método de Diffie-Hellman.

Hidden page

Hidden page

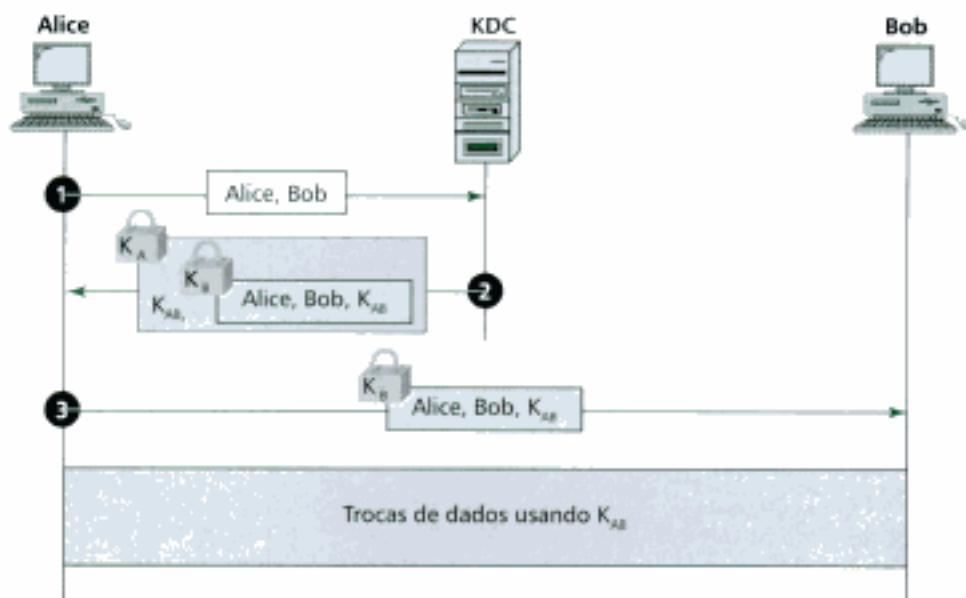


Figura 30.13: Primeira abordagem usando o KDC.

Eve pode atacar usando o *replay* discutido anteriormente. Ela pode salvar a mensagem na etapa 3, assim como as mensagens de dados, e responder a todos.

Protocolo Needham-Schroeder Outra abordagem é o **protocolo Needham-Schroeder**. Esse protocolo é muito elegante e serve como base para outros tantos. A idéia desse protocolo é usar um mecanismo de desafio/resposta entre as partes para criar um protocolo livre de fraquezas. Na última versão desse protocolo, Needham e Schroeder usaram quatro *nonces* diferentes: R_A , R_B , R_1 e R_2 . A Figura 30.14 mostra as sete etapas deste protocolo.

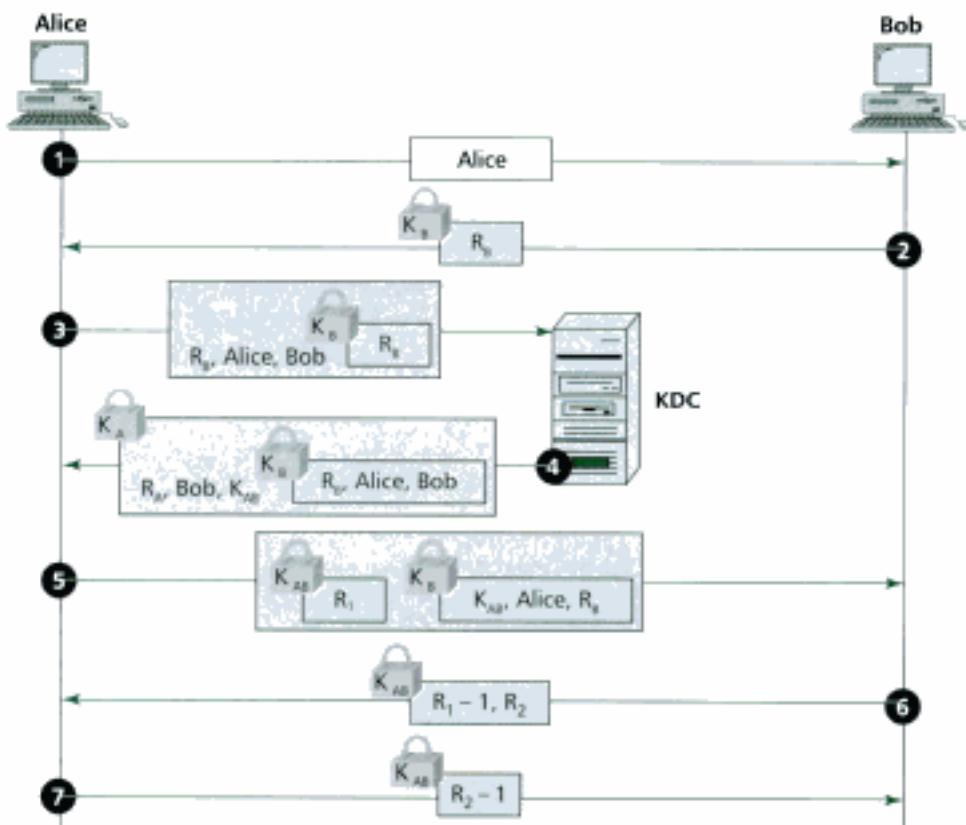


Figura 30.14: Protocolo de Needham-Schroeder.

A seguir temos a descrição abreviada de cada etapa:

Etapa 1 Alice envia sua identidade para o Bob e, por meio dela, declara a vontade de estabelecer uma comunicação com ele.

Etapa 2 Bob usa a *nonce* R_B e cifra a mensagem usando a chave simétrica K_B . A *nonce* R_B foi planejada para ser enviada para o KDC, mas ela foi enviada para Alice. Alice retransmite o K_B ao KDC para provar que ela é a pessoa que se comunicou com o Bob são na verdade a mesma pessoa, e não a Eve que está tentando se comunicar com o KDC.

Etapa 3 Alice envia uma mensagem ao KDC que inclui a *nonce* (R_A) e a identidade dela, a identidade do Bob e a *nonce* cifrada do Bob.

Etapa 4 O KDC envia uma mensagem cifrada para Alice que inclui a *nonce* da Alice, a identidade do Bob, a chave de sessão e um *ticket* cifrado para o Bob que inclui a *nonce* dele. Nesse caso, Alice recebeu a resposta da sua *nonce* de desafio e a chave de sessão.

Etapa 5 Alice envia o *ticket* do Bob juntamente com a nova *nonce* R_1 para desafiá-lo.

Etapa 6 Bob responde ao desafio de Alice e a desafia (R_2). Observe que a resposta ao desafio de Alice é o valor $(R_1)^{-1}$. Isto assegura que Bob decifrou o R_1 cifrado na etapa anterior. Em outras palavras, a nova cifragem garante que não foi Eve quem enviou a mesma mensagem cifrada de volta.

Etapa 7 Alice responde ao desafio de Bob. Outra vez, observe que a resposta transporta $(R_2)^{-1}$ e não R_2 .

Protocolo de Otway-Rees Uma terceira abordagem usa outro protocolo elegante, o **protocolo de Otway-Rees**. Esse protocolo tem a vantagem de possuir menos etapas que o protocolo de Needham-Schroeder. A Figura 30.15 ilustra este protocolo em cinco etapas.

Etapa 1 Alice envia uma mensagem ao Bob que inclui a *nonce* R (comum aos dois), as identidades de Alice e Bob; além disso, envia um *ticket* para o KDC que inclui a *nonce* R_A de Alice (o desafio para o KDC usar), uma cópia do *nonce* comum (R) e as identidades de Alice e Bob.

Etapa 2 Bob cria o mesmo tipo de *ticket*, mas composto pela própria *nonce* dele (R_B); ambos os *tickets* são enviados ao KDC.

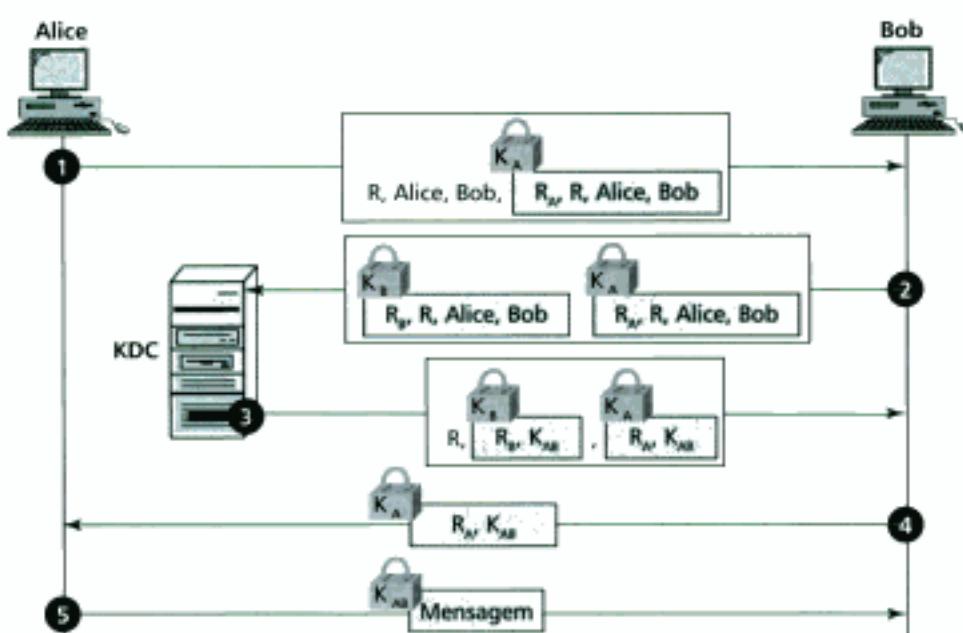


Figura 30.15 Protocolo de Otway-Rees.

Hidden page

tificado de forma estruturada. Ele usa um protocolo bem conhecido denominado ASN-1 (Abstract Syntax Notation 1) que define os campos de uma maneira semelhante ao que fazem os programadores em C.

Não iremos detalhar o ASN-1 aqui, mas listamos na Tabela 30.1 alguns dos campos definidos pelo X.509 e seus respectivos significados.

TABELA 30.1 Campos X.509

Campo	Explicação
Versão	Número da versão do X.509
Número serial	Único identificador usado pelo CA
Assinatura	Assinatura do certificado
Emissor	O nome da CA definida pelo X.509
Validade	Período de validade do certificado
Nome	Nome da entidade cuja chave pública está sendo certificada
Chave pública	A chave pública e os algoritmos que ela usa

Infra-estrutura de Chave Pública (PKI)

Quando queremos usar chaves públicas universalmente chegamos a um problema bastante parecido ao encontrado no DNS (Domain Name System), Capítulo 25. Vimos que não podíamos confiar apenas em um servidor DNS para responder todas as consultas. Concluímos que era necessário muitos servidores. Assim, vimos que a melhor solução era organizar os servidores formando hierarquias. Se Alice precisar do endereço IP do Bob, Alice envia uma mensagem para o servidor local que pode conhecer ou não o endereço IP do Bob. O servidor local consulta um servidor pai hierarquicamente acima, caso necessário, até que o endereço IP seja determinado.

Do mesmo modo, uma solução para as consultas de chave pública é denominada **infra-estrutura de chave pública** (Public-Key Infrastructure – PKI). A Figura 30.16 mostra um exemplo dessa hierarquia.

No primeiro nível, temos a autoridade CA raiz que certifica a *performance* das CAs no segundo nível. Estas CAs de nível-1 podem operar em uma área geográfica ou logicamente grande. As CAs do nível-2 podem operar em áreas geográficas menores.

Nesta hierarquia, todos confiam na CA raiz, mas as pessoas podem confiar ou não nas CAs intermediárias. Se Alice precisar obter o certificado do Bob, ela pode determinar uma CA em algum lugar mais próximo e solicitar a emissão do certificado. Contudo, Alice pode não confiar na CA. Nesse caso, Alice pode subir no hierarquia tentando chegar em uma autoridade em quem ela realmente confia. A pesquisa pode culminar na CA raiz.

A estrutura PKI é uma novidade na Internet. Indubitavelmente, a importância dada a PKI aumentará nos próximos anos.

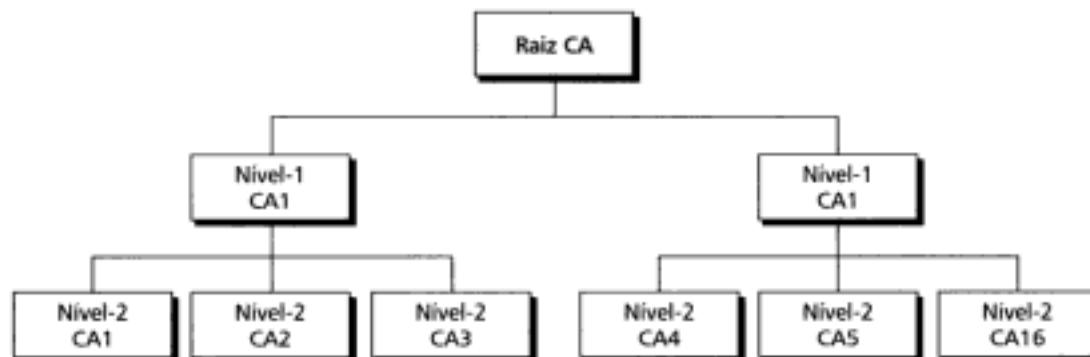


Figura 30.16 Hierarquia PKI.

Hidden page

Operação

Um processo cliente (Alice) recebe um serviço do processo rodando no servidor real (Bob) em seis etapas, conforme ilustra a Figura 30.18.

Etapa 1

Alice envia a solicitação dela ao AS na forma de texto limpo, usando a identidade registrada dela.

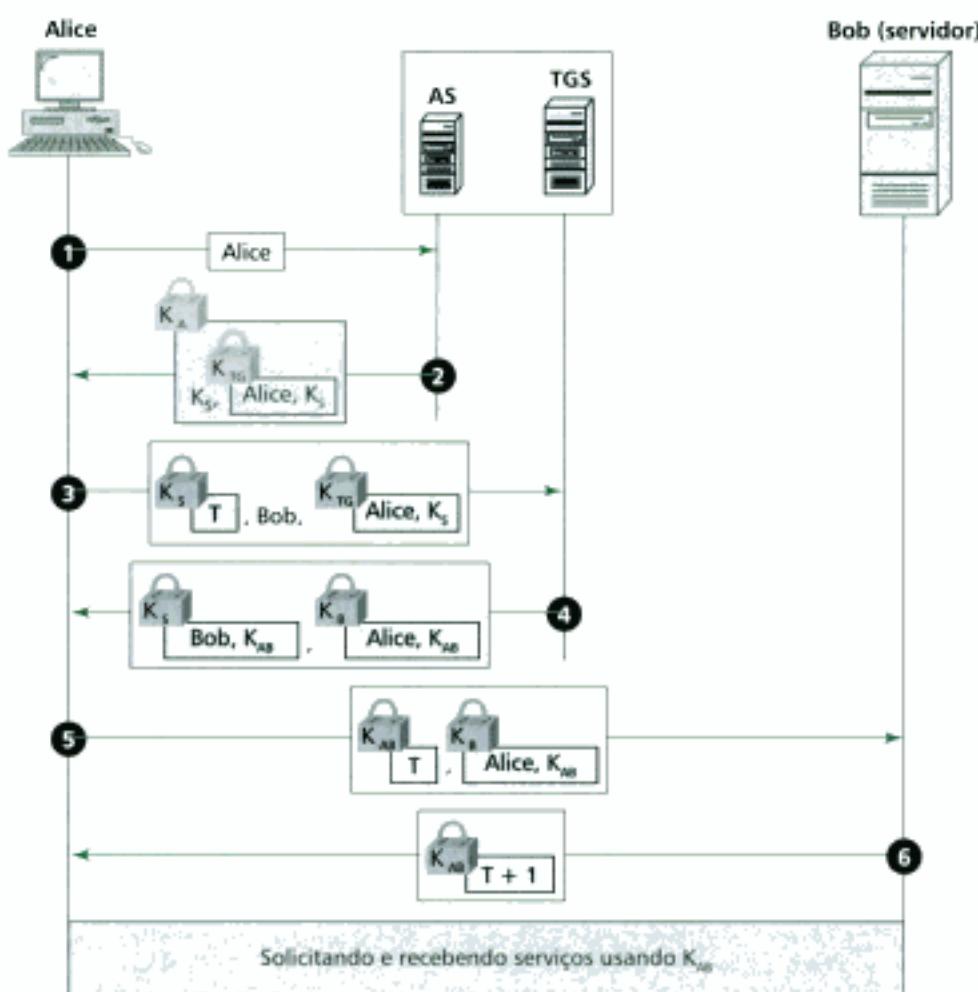


Figura 30.18 Exemplo Kerberos.

Etapa 2

O AS envia para a Alice uma mensagem cifrada com a chave simétrica dela (K_A). A mensagem possui dois itens: uma chave de sessão (K_S) que é utilizada pela Alice para contatar o TGS e um *ticket* para o servidor TGS que está cifrado através da chave simétrica TGS (K_{TG}). Alice desconhece a chave K_{TG} , mas quando a mensagem chega, ela digita a senha. Juntos, a senha e o algoritmo apropriado criam a chave K_A , se a senha estiver correta. Em seguida, a senha é destruída imediatamente; assim, ela não é transmitida pela rede e nem sequer é armazenada no terminal. Ela é utilizada somente para criar a chave K_A . Agora, o processo usa K_A para decifrar a mensagem enviada; K_S e o *ticket* são extraídos da mensagem.

Etapa 3

Alice pode enviar três itens ao servidor TGS. O primeiro é o *ticket* recebido do AS. O segundo é o nome do servidor real (Bob) e o terceiro é uma informação de *timestamp* cifrada pela chave K_S . A informação de *timestamp* evita que Eve ataque fazendo um *replay*.

Hidden page

Hidden page

Hidden page

22. Em um ataque _____, um intruso fica à espreita entre duas partes que estão se comunicando, interceptando e replicando as mensagens delas.
- Return*
 - Man-in-the-middle*
 - Bucket-in-the-middle*
 - Replay*
23. Um _____ é um terceiro conflável que estabelece uma chave simétrica entre as duas partes que desejam se comunicar com segurança.
- KDC
 - CA
 - PKI
 - TGS
24. No protocolo _____, um número denominado *nonce* é decrementado de 1 tal que o intruso não pode enviar a mesma mensagem uma segunda vez.
- Diffie-Hellman
 - Needham-Schroeder
 - Otaway-Rees
 - Kerberos
25. O _____ é um protocolo de autenticação que necessita de um servidor de autenticação (AS) e um servidor de *tickets* (TGS).
- Diffie-Hellman
 - Needham-Schroeder
 - Otaway-Rees
 - Kerberos
26. O _____ é o KDC no protocolo Kerberos.
- AS
 - TGS
 - Servidor real
 - Servidor de dados
27. O _____ emite *tickets* para o servidor real.
- AS
 - TGS
 - Servidor real
 - Servidor de dados
28. Na criptografia com chave _____, o acesso a todas as chaves, de todas as pessoas, é público.
- Privada
 - Simétrica
 - Pública
 - Certificada
29. Um protocolo denominado _____ descreve o tipo de certificado emitido por uma CA de forma estruturada.
- X.509
 - CA nível-1
 - KDC
 - Kerberos
30. O Windows 2000 usa um protocolo de autenticação denominado _____.
- Diffie-Hellman
 - Needham-Schroeder
 - Otway-Rees
 - Kerberos

Exercícios

31. Adicione um nível de cifragem/decifram de chave simétrica à Figura 30.4 para fornecer privacidade.
32. Adicione um nível de cifragem/decifram de chave pública à Figura 30.4 para fornecer privacidade.
33. Mostre que G^y é igual a $(G^x)^y$. Use $G = 11$, $x = 3$ e $y = 4$.
34. Prove que o resultado de $G^y \bmod N$ é igual ao resultado $(G^x \bmod N)^y \bmod N$. Use $G = 7$, $x = 2$, $y = 3$ e $N = 11$.
35. O fato do resultado de $G^y \bmod N$ ser idêntico ao resultado $(G^x \bmod N)^y \bmod N$ pode simplificar tremendamente o cálculo de $G^y \bmod N$. Use este fato para determinar $7^{18} \bmod 11$.
- Sugestão:* fatore o 18 e realize três cálculos.
36. Qual é o valor da chave simétrica no protocolo de Diffie-Hellman se $G = 7$, $N = 23$, $x = 3$ e $y = 57$
37. Quais são os valores de R_1 e R_2 no protocolo de Diffie-Hellman se $G = 7$, $N = 23$, $x = 3$ e $y = 57$
38. No protocolo de Diffie-Hellman, o que acontece se x e y tiverem os mesmos valores, isto é, se acidentalmente a Alice e o Bob escolheram os mesmos números? Os valores de R_1 e R_2 são os mesmos? O valor da chave de sessão é calculado através da chave de Alice e de Bob? Use um exemplo para comprovar suas respostas.
39. Qual(is) dos números a seguir é um bom candidato para N no protocolo de Diffie-Hellman? 7, 11, 15, 21, 33, 37 ou 47.

40. Na Figura 30.13, (primeira abordagem usando KDC), o que acontece se o *ticket* do Bob não for cifrado na etapa 2 com K_b , mas com K_{AB} na etapa 3?
41. Por que há necessidade de quatro números *nonces* no protocolo de Needham-Schroeder?
42. No protocolo de Needham-Schroeder, de que forma Alice é autenticada pelo KDC? E o Bob? Como o KDC é autenticado pela Alice? E pelo Bob?
43. Você pode explicar por que no protocolo de Needham-Schroeder, Alice é a parte que mantém contato com o KDC, mas no protocolo Otway-Rees é o Bob que está em contato com o KDC?
44. Há quatro números *nonces* (R_A , R_B , R_1 e R_2) no protocolo de Needham-Schroeder, mas somente três (R_A , R_B , R_1) no protocolo de Otway-Rees. Você pode explicar por que é necessário um *nonce* adicional (R_2) no primeiro protocolo?
45. Por que precisamos somente de um *timestamp* no Kerberos em vez dos quatro números *nonces* do protocolo de Needham-Schroeder ou três no protocolo Otway-Rees?
46. Na abordagem bidirecional para autenticação (mostrada na Figura 30.10), supondo que são permitidas múltiplas sessões de autenticação, Eve intercepta o *nonce* R_B de Bob (na segunda sessão) e transmite como o *nonce* de Alice da segunda sessão. Bob, sem verificar que este número *nonce* é o mesmo que ele tinha enviado, cifra R_B e o coloca na mensagem com esse *nonce*. Eve usa a R_B cifrada e finge ser Alice, continuando a primeira sessão e respondendo com o R_B cifrado. Isto é conhecido como ataque por reflexão. Apresente as etapas deste cenário.

Protocolos de Segurança na Internet

Todos os princípios e conceitos discutidos nos dois capítulos anteriores serão utilizados para fornecer os aspectos de segurança essenciais ao modelo da Internet. Em particular, serão aplicadas medidas de segurança nas camadas de rede, transporte e de aplicação.

A implementação das medidas de segurança na camada IP é bastante complexa, especialmente porque cada dispositivo dessa camada deve estar habilitado para prover segurança. O protocolo IP fornece serviços não somente para as aplicações do usuário, mas para outros protocolos também, tal como OSPF, ICMP e IGMP. Assim, a implementação de segurança nesse nível não é muito efetiva, a menos que todos os dispositivos estejam preparados para utilizá-la. Discutiremos um protocolo denominado IPSec cuja finalidade é fornecer segurança na camada IP.

Na camada de transporte a implementação da segurança é ainda mais complicada. Poderíamos modificar a aplicação ou modificar a camada de transporte para agregar segurança. Em vez disso, estudaremos um protocolo que "cola" uma nova camada no nível de transporte para fornecer segurança a favor dessa camada.

Na camada de aplicação cada aplicação é responsável em fornecer seu próprio nível de segurança. A implementação de segurança nesse nível é a mais simples de todas. Ela diz respeito a duas entidades: o cliente e o servidor. Analisaremos um método de segurança na camada de aplicação denominado PGP.

Um mecanismo usado freqüentemente para manter a integridade de uma empresa ou organização é um *firewall*. Faremos uma breve exposição sobre *firewalls* neste capítulo.

Finalmente, visto que este é o último capítulo sobre segurança, analisaremos uma tecnologia interessante e atual, as VPNs (Virtual Private Networks), que usam a Internet pública como meio de transmissão, mas possui um nível de segurança característico de uma rede privada.

31.1 SEGURANÇA NA CAMADA DE REDE: IPSec

O **IP Security (IPSec)** é uma coleção de protocolos desenvolvidos pelo IETF (Internet Engineering Task Force) para fornecer segurança para um pacote da camada IP. O IPSec não define o uso de nenhuma técnica de cifragem ou método de autenticação. Na verdade, ele fornece uma estrutura e um mecanismo; ele deixa a escolha do tipo de cifragem, autenticação e métodos de *hashing* para o usuário.

Security Association

O IPSec requer uma conexão lógica entre dois *hosts* usando um *protocolo de sinalização*, chamado **Security Association (SA)**. Em outras palavras, o IPSec precisa que o protocolo sem conexão IP seja modificado para um protocolo orientado à conexão antes da segurança ser implementada efetivamente. Uma conexão SA é uma conexão *simplex* (unidirecional) entre a fonte e o destino. Assim, se houver a necessidade de estabelecermos conexão *full-duplex* (bidirecional) serão necessárias duas conexões SA, uma em cada direção. Uma conexão SA é definida unicamente através de três elementos:

1. Um SPI (Security Parameter Index) de 32 bits *age* como um identificador de circuito virtual em protocolos orientados à conexão, como o Frame Relay e o ATM.
2. O tipo de protocolo usado para a segurança. Veremos em breve que o IPSec define dois protocolos alternativos: AH e ESP.
3. A origem do endereço IP.

Dois Modos de Operação

O IPSec opera em dois modos diferentes, a saber: modo de transporte e modo de tunelamento. O modo define aonde o cabeçalho IPSec será adicionado ao pacote IP.

Modo de Transporte

Neste modo, o cabeçalho IPSec é adicionado entre o cabeçalho IP e o restante do pacote, conforme ilustra a Figura 31.1

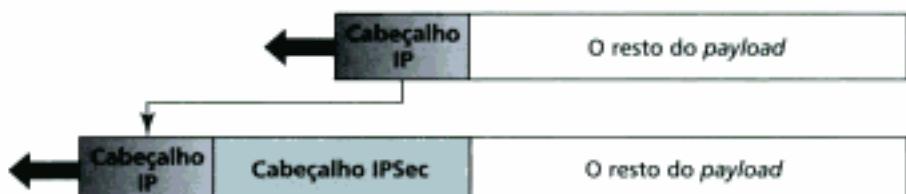


Figura 31.1 Modo de transporte.

Modo de Tunelamento

Neste modo, o cabeçalho IPSec é colocado logo à frente do cabeçalho IP original. Um novo cabeçalho IP é adicionado na frente do cabeçalho IPSec. O cabeçalho IPSec, o cabeçalho IP original e o restante do pacote são tratados como o *payload* no modo de tunelamento. A Figura 31.2 mostra o pacote IP original e o novo pacote IP.

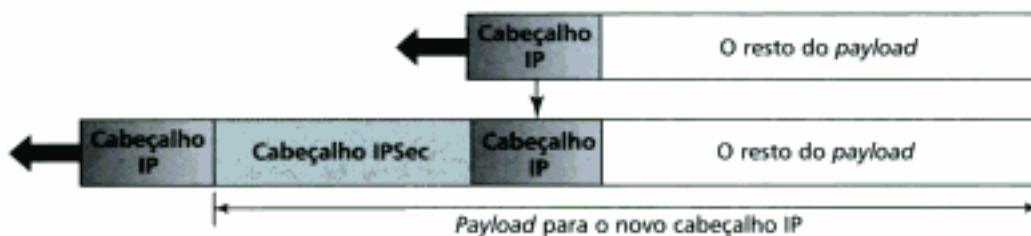


Figura 31.2 Modo de tunelamento.

Dois Protocolos de Segurança

O IPSec define dois protocolos de segurança: Authentication Header (AH) e Encapsulating Security Payload (ESP). Analisaremos esses dois protocolos a seguir.

AH

O protocolo AH (Authentication Header) foi desenvolvido para autenticar o *host* de origem e assegurar a integridade do *payload* transportado pelo pacote IP dele. O protocolo calcula a síntese da mensagem, usando uma função de *hashing* e uma chave simétrica, e insere a síntese no cabeçalho do protocolo AH. O AH é colocado na posição correta no datagrama IP baseado no modo de operação (transporte ou tunelamento). A Figura 31.3 ilustra os campos e a posição do protocolo AH no modo de transporte.

Quando um datagrama IP transporta um AH, o valor original no campo protocolo do cabeçalho IP é substituído pelo valor 51. Um campo interno ao protocolo AH (campo próximo cabeçalho) define o valor original do campo protocolo (o tipo de *payload* sendo transportado pelo datagrama IP). Para agregar o protocolo AH ao datagrama IP é necessário seguir as seguintes etapas:

1. O protocolo AH é adicionado ao campo *payload* juntamente com o campo de dados autenticado todo configurado em zero.
2. Pode ser necessário adicionar *padding* (*bits* de enchimento) para tornar o tamanho total do pacote par, caso algum algoritmo de *hashing* particular exija isso.
3. A função de *hashing* é baseada no tamanho do pacote total. Entretanto, somente os campos do cabeçalho IP que não sofreram modificação durante a transmissão serão incluídos no cálculo da síntese da mensagem (autenticação dos dados).
4. A autenticação dos dados está incluída no protocolo AH.
5. O cabeçalho IP é adicionado ao pacote após a troca do valor do campo protocolo para 51.

A seguir temos uma breve descrição de cada campo:

- **Próximo Cabeçalho.** Este campo possui 8 *bits* e define o tipo de *payload* transportado no datagrama IP (TCP, UDP, ICMP, OSPF e assim por diante). A finalidade desse campo é a mesma do campo protocolo no cabeçalho IP. Em outras palavras, o processo copia o valor do campo protocolo do datagrama IP para este campo. O valor do campo protocolo no datagrama IP é trocado para 51 para mostrar que o pacote transporta o protocolo AH.
- **Tamanho do Payload.** O campo tamanho do *payload* de 8 *bits* é enganoso. Ele não define o tamanho verdadeiro do *payload* em si. De fato, é definido apenas o tamanho do protocolo AH, em múltiplo de 4 *bytes*, sem incluir os 8 primeiros *bytes*.

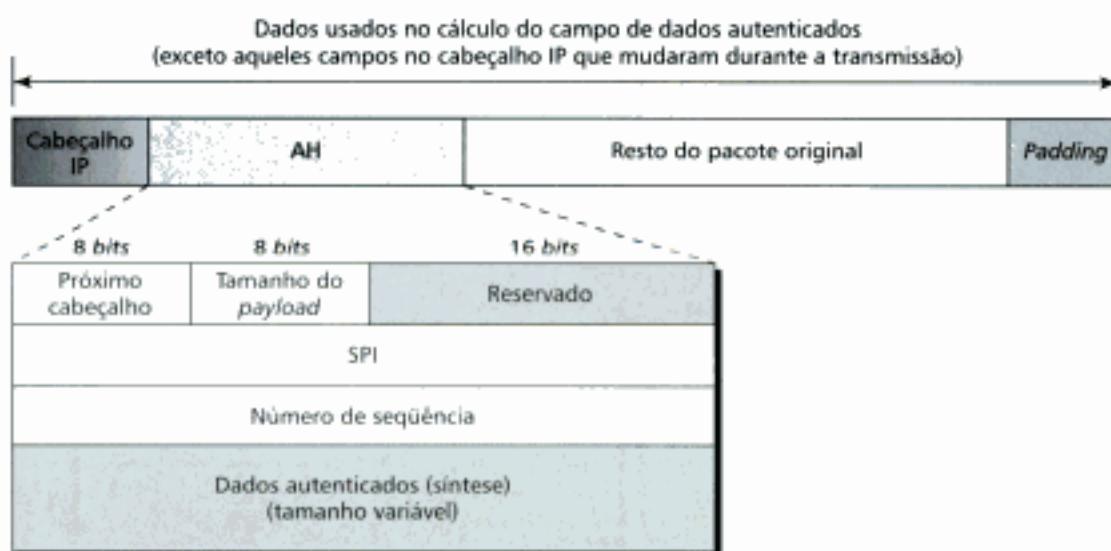


Figura 31.3 AH.

- **SPI (Security Parameter Index).** Este campo tem 32 bits de comprimento e desempenha um papel crucial como identificador do circuito virtual (VCI), mantendo-se fixo para todos os pacotes durante uma conexão SA.
- **Número de Seqüência.** Um número de seqüência fornece a informação sobre a ordem de seqüência dos datagramas IP. Um número de seqüência evita a repetição dos datagramas. Observe que o número de seqüência não é repetido até mesmo quando um pacote é retransmitido.
- **Dados Autenticados.** Finalmente, o campo dados autenticados é o resultado da aplicação de uma função de *hashing* ao datagrama IP como um todo, exceto os campos que não sofreram modificação durante o trânsito (p. ex., *time-to-live*).

O protocolo AH é uma fonte de autenticação e integridade de dados, mas não oferece privacidade.

Protocolo ESP

O protocolo AH não suporta privacidade. A versão IPSec posterior definiu um protocolo **ESP (Encapsulating Security Payload)** alternativo fornecendo autenticação, integridade e privacidade da informação. O ESP adicionou um cabeçalho e um rótulo (*trailer*) ao datagrama IP original. Observe que os dados autenticados via ESP são adicionados ao final do pacote, o que torna o seu cálculo mais simples. A Figura 31.4 mostra a localização do cabeçalho e *trailer* ESP.

Quando um datagrama IP transporta um cabeçalho e um *trailer* ESP, o valor do campo protocolo no cabeçalho IP é modificado para 50. Um campo interno ao *trailer* ESP (o campo próximo cabeçalho) mantém o valor original do campo protocolo contendo tipo de *payload* transportado pelo datagrama IP, tal como TCP e o UDP. O procedimento ESP segue as seguintes etapas:

1. Um *trailer* ESP é adicionado ao *payload*.
2. O *payload* e o *trailer* são criptografados.
3. O cabeçalho ESP é adicionado ao *payload*.
4. O cabeçalho ESP, o *payload* restante e um *trailer* ESP são usados para gerar dados autenticados.
5. Os dados autenticados são agregados ao final do *trailer* ESP.
6. O cabeçalho IP é adicionado após a mudança do valor do protocolo para 50.

Os campos internos ao cabeçalho e *trailer* são os seguintes:

- **SPI.** Este campo de 32 bits é similar ao campo SPI definido no protocolo AH.



Figura 31.4 ESP.

- **Número de Seqüência.** Este campo de 32 bits também é similar ao campo número de seqüência do protocolo AH.
- **Bits de Enchimento (Padding).** Este campo possui um tamanho variável de 0s (0 a 255 bytes) servindo como *padding*.
- **Tamanho do Pad.** Este campo de 8 bits define a quantidade de bytes de padding. O valor geralmente situa-se entre 0 e 255, sendo que o uso do valor máximo é raro.
- **Próximo Cabeçalho.** Este campo de 8 bits é bastante semelhante ao campo próximo cabeçalho definido no protocolo AH. Esse campo tem a mesma finalidade do campo protocolo no cabeçalho IP (antes do encapsulamento).
- **Dados Autenticados.** Finalmente, o campo dados autenticados é o resultado da aplicação do esquema de autenticação em todas as partes do datagrama. Observe que existe diferença entre os dados autenticados no AH e no ESP. No AH, parte do cabeçalho IP é incluído no cálculo dos dados autenticados; no ESP isso não acontece.

O protocolo ESP fornece autenticação, integridade e privacidade à informação.

IPv4 e IPv6

IPSec suporta tanto a versão IPv4 quanto a versão IPv6. Entretanto, na versão IPv6 ambos protocolos (AH e ESP) são parte do cabeçalho estendido.

AH versus ESP

O protocolo ESP foi desenvolvido depois do protocolo AH estar em pleno uso. O ESP tem tudo que o protocolo AH incorpora, agregando mais uma funcionalidade: a privacidade. Então, a questão é: Por que precisamos do protocolo AH? A resposta é que não precisamos. Contudo, o AH está implementado em alguns produtos comerciais na Internet. Isso significa que ele ainda fará parte das aplicações da Internet por algum tempo.

31.2 SEGURANÇA NA CAMADA DE TRANSPORTE

A **TLS (Transport Layer Security)** foi desenvolvida para fornecer segurança à camada de transporte. O protocolo TLS foi derivado de um protocolo de segurança denominado Secure Sockets Layer (SSL), criado pela Netscape para fornecer segurança a WWW. Sendo assim, podemos dizer que o TLS é uma versão sem dono do SSL, desenvolvido pela IETF. Para realizar transações na Internet, um *browser* precisa do seguinte:

1. Um usuário comprando pela Internet precisa ter certeza de que o servidor onde está hospedada a aplicação de venda realmente pertence à empresa que lhe está vendendo os produtos e não a um impostor. Por exemplo, um consumidor deve questionar o vendedor sobre autenticidade dele antes de enviar o seu número do cartão de crédito. Em outras palavras, o servidor deve ser autenticado.
2. O consumidor precisa ter certeza de que o conteúdo da mensagem não foi modificado durante uma transmissão. Uma conta de R\$100,00 não pode virar R\$1000,00 repentinamente. A integridade da mensagem deve ser preservada.
3. O consumidor precisa ter certeza de que um impostor não irá interceptar informação delicada, por exemplo, contendo o número do cartão de crédito.

Existem outros aspectos de segurança opcionais que podem ser agregadas à lista acima. Por exemplo, a empresa que efetua a venda pode querer autenticar um consumidor. O TLS suporta características adicionais que oferecem estes aspectos de segurança.

Hidden page

Hidden page

Hidden page

Filtros de Pacotes

A aplicação mais importante de um *firewall* é a filtragem de pacotes. Um *firewall* pode atacar ou bloquear os pacotes baseados na informação contida nos cabeçalhos dos pacotes das camadas de rede e/ou de aplicação: endereço IP de origem e de destino, número de porta de origem e de destino, tipo de protocolo (TCP ou UDP) e muito mais.

A implementação do *firewall* como filtro de pacotes é feita nos roteadores da rede que usam uma tabela de filtragem para tomar decisão sobre o descarte ou não de pacotes. A Figura 31.10 é um exemplo de tabela de filtragem desse tipo de *firewall*.

De acordo com a figura, os seguintes pacotes são filtrados:

1. Pacotes oriundos da rede 131.34.0.0 são bloqueados por precaução. Observe que o * significa qualquer pacote.
2. Pacotes destinados a qualquer servidor de TELNET interno (porta 23) são bloqueados.
3. Pacotes destinados ao *host* interno 194.78.20.8 são bloqueados. A organização deseja que este *host* seja de uso exclusivamente interno.
4. Pacotes de saída destinados a quaisquer servidores de HTTP externos (porta 80) são bloqueados. A organização não quer que seus funcionários tenham acesso à Internet.

Um *firewall* como filtro de pacotes seleciona os pacotes baseado na informação das camadas de rede e de transporte.

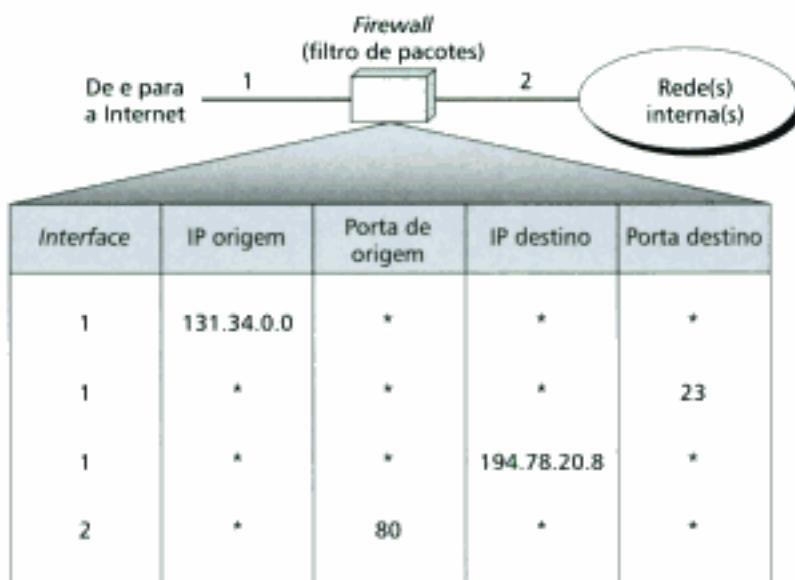


Figura 31.10 Filtragem de pacotes.

Proxy Firewall

Um filtro de pacotes baseia-se na informação disponível nos cabeçalhos dos pacotes das camadas de rede e de transporte (IP e TCP/UDP) para tomar decisões sobre o descarte ou não de pacotes. Entretanto, é comum querermos filtrar uma mensagem com base no conteúdo da mensagem em si (camada de aplicação). Por exemplo, imagine que uma empresa deseja implementar a seguinte política de acesso às suas páginas na Web: somente aqueles usuários da Internet que têm estabelecido alguma relação de negócio com a empresa podem ter acesso ao conteúdo das páginas; o acesso aos demais usuários é sumariamente bloqueado. Nesse caso, o *firewall* implementado como filtro de pacotes não é adequado porque ele não pode, por exemplo, distinguir entre os diferentes tipos de pacotes recebidos na porta TCP número 80 (HTTP). O acesso ou não deve ser feito baseado na informação contida na camada de aplicação (por exemplo, nos URLs).

Uma solução interessante é instalar um servidor de *proxy* (às vezes denominado *gateway*) que fica situado entre o computador do usuário cliente e os computadores da corporação. Quando o processo cliente do usuário envia uma mensagem, o **servidor de proxy** roda um processo servidor que recebe a solicitação. O servidor abre o pacote no nível da camada de aplicação, verifica o conteúdo dele e o encaminha, ou não, baseado na legitimidade da informação. Se houver aprovação para encaminhamento, o servidor age como um processo cliente e envia a mensagem para o servidor real, isto é, o servidor que deve atender a solicitação na corporação. Se não houver aprovação, a mensagem é destruída e uma mensagem de erro é transmitida ao usuário externo. Desse modo, as solicitações dos usuários externos são filtradas com base em critérios estabelecidos na camada de aplicação. A Figura 31.11 mostra uma implementação utilizando o servidor de *proxy* (*firewall*).

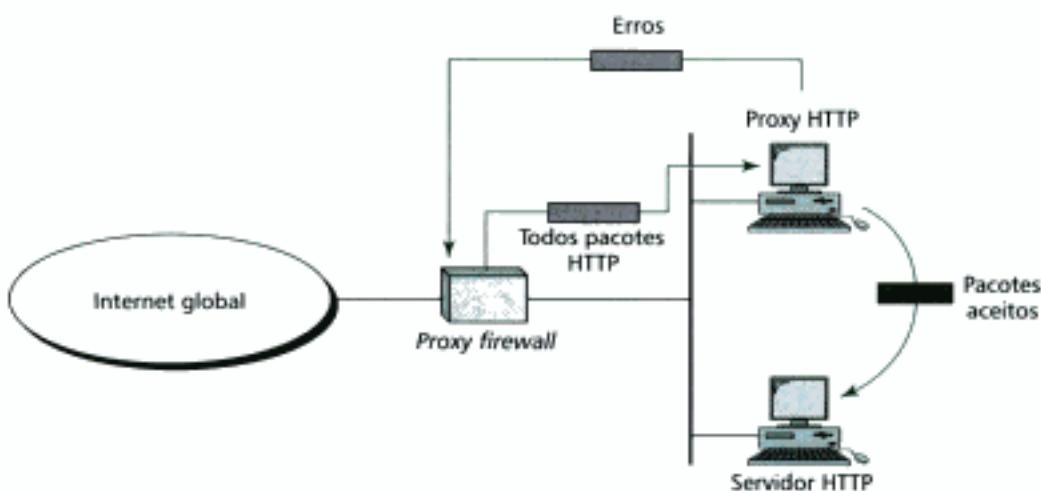


Figura 31.11 *Proxy firewall*.

Um servidor de *proxy* (*firewall*) filtra na camada de aplicação.

31.5 VIRTUAL PRIVATE NETWORK

Virtual Private Network (VPN) é uma tecnologia que ultimamente tem sido bastante procurada pelas grandes organizações que usam a Internet global tanto para a comunicação privativa interna, entre as unidades da empresa, quanto para a comunicação privativa externa.

Redes Privadas

Uma rede privada foi desenvolvida para ser utilizada dentro de uma organização. Ela permite acessar os recursos compartilhados e, ao mesmo tempo, fornecer privacidade. Antes de tratarmos alguns aspectos das redes privadas, vamos definir dois termos correlacionados: *intranet* e *extranet*.

Intranet

Uma **intranet** é uma rede privada (LAN) que se utiliza do modelo da Internet. Contudo, o acesso aos recursos fica limitado aos usuários internos a uma organização. A rede usa programas aplicativos definidos para a Internet global, tal como o HTTP e hospeda servidores Web, servidores de impressão, servidores de arquivos e assim por diante.

Extranet

Uma **extranet** é essencialmente a mesma coisa que uma intranet, exceto pelo fato de que alguns recursos podem ser acessados por grupos específicos de usuários externos a uma organização, sob o controle do administrador da rede. Por exemplo, uma organização pode permitir aos consumido-

Hidden page

Hidden page

Hidden page

Hidden page

Questões de Múltipla Escolha

16. O IPSec requer uma conexão lógica entre dois *hosts* usando um protocolo de sinalização denominado _____.
 a. AH
 b. SA
 c. PGP
 d. TLS
17. O protocolo de *handshake* e de troca de dados são partes do _____.
 a. CA
 b. KDC
 c. TLS
 d. SSH
18. _____ é uma coleção de protocolos que agregam segurança na camada de rede (IP).
 a. TLS
 b. SSH
 c. PGP
 d. IPSec
19. _____ é um protocolo de segurança da camada de rede que fornece somente integridade e autenticação de usuários.
 a. AH
 b. PGP
 c. ESP
 d. IPSec
20. _____ é um protocolo de segurança da camada de rede que fornece somente integridade, autenticação de usuários e privacidade.
 a. AH
 b. PGP
 c. ESP
 d. IPSec
21. Um datagrama IP transporta um cabeçalho de autenticação se o campo _____ do cabeçalho IP tiver o valor 51.
 a. Próximo cabeçalho
 b. Protocolo
 c. SPI
 d. Número de seqüência
22. Um (a) _____ pode encaminhar ou bloquear pacotes baseados na informação dos cabeçalhos dos datagramas IP ou TCP/UDP.
 a. Proxy
 b. Filtro de pacotes
 c. Síntese da mensagem
- d. Chave privada
23. O campo _____ no cabeçalho de autenticação e o cabeçalho ESP definem o método de segurança usado na geração de dados autenticados.
 a. Padding
 b. Número de seqüência
 c. Dados autenticados
 d. SPI
24. _____ é um protocolo de segurança da camada de transporte.
 a. TLS
 b. PGP
 c. IPSec
 d. AH
25. Um método para fornecer transporte seguro de *e-mails* é denominado _____.
 a. TLS
 b. SA
 c. PGP
 d. IPSec
26. Um servidor _____ pode encaminhar ou bloquear mensagens baseado na informação contida na mensagem.
 a. Proxy
 b. De filtragem de pacotes
 c. De síntese da mensagem
 d. De chave privada
27. Uma rede _____ está totalmente isolada da Internet global.
 a. Privada
 b. Híbrida
 c. Virtual privada
 d. Todas as alternativas anteriores
28. Uma rede _____ utiliza uma linha dedicada para fornecer comunicação intra-organização e usa a Internet para fornecer comunicação inter-organização.
 a. Privada
 b. Híbrida
 c. Virtual privada
 d. Todas as alternativas anteriores
29. Uma rede VPN usa _____ para dar garantias de privacidade.
 a. IPSec
 b. Tunelamento
 c. (a) e (b)
 d. Nenhuma das alternativas anteriores

30. Numa rede VPN, _____ é (são) cifrado(s).
 a. O datagrama interno
 b. O datagrama externo
 c. Tanto o datagrama interno quanto o datagrama externo
 d. Nenhum datagrama
31. O tunelamento é uma técnica na qual o datagrama IP primeiramente é _____ e, em seguida, _____.
 a. Encapsulado em outro datagrama; cifrado
 b. Cifrado; encapsulado em outro datagrama
 c. Autenticado; cifrado
32. Uma rede _____ é uma rede privada sem nenhum acesso externo que usa os protocolos TCP/IP.
 a. Internet
 b. internet
 c. Intranet
 d. Extranet
33. Uma _____ é uma rede privada com acesso externo limitado que usa os protocolos TCP/IP.
 a. Internet
 b. internet
 c. Intranet
 d. Extranet

Exercícios

34. Apresente os valores dos campos AH na Figura 13.3. Assuma que o campo de dados autenticados usa somente 128 bits.
35. Apresente os valores dos campos cabeçalho e trailer ESP na figura 31.4.
36. Redesenhe a Figura 31.3 se o protocolo AH for utilizado no modo de tunelamento.
37. Redesenhe a Figura 31.4 se o protocolo ESP for utilizado no modo de tunelamento.
38. Desenhe uma figura mostrando a posição do protocolo AH no IPv6.
39. Desenhe uma figura mostrando a posição do protocolo ESP no IPv6.
40. Compare o protocolo de *handshake* na Figura 31.6 com os protocolos de autenticação discutidos no Capítulo 30. Você pode dizer que protocolo no Capítulo 30 é semelhante ao protocolo de *handshake*?
41. O protocolo PGP na Figura 31.7 usa três chaves. Explique cada uma delas.
42. O protocolo PGP precisa dos serviços de um KDC? Explique sua resposta.
43. O protocolo PGP precisa dos serviços de uma CA? Explique sua resposta.
44. Uma VPN pode utilizar o IPSec no modo de transporte? Explique sua resposta.

Hidden page

TABELA A.1 Tabela ASCII (continuação)

Decimal	Hexadecimal	Binário	Caractere	Descrição
20	14	0010100	DC4	Controle de dispositivo 4
21	15	0010101	NAK	Confirmação negativa
22	16	0010110	SYN	Síncronismo
23	17	0010111	ETB	Fim do bloco de transmissão
23	18	0011000	CAN	Cancela
25	19	0011001	EM	Fim de meio de transmissão
26	1A	0011010	SUB	Substitui
27	1B	0011011	ESC	Escape
28	1C	0011100	FS	Separador de arquivos
29	1D	0011101	GS	Separador de grupos
30	1E	0011110	RS	Separador de registros
31	1F	0011111	US	Separador de unidades
32	20	0100000	SP	Espaço
33	21	0100001	!	Sinal de exclamação
34	22	0100010	"	Aspas
35	23	0100011	#	Símbolo de cardinal
36	24	0100100	\$	Críptico
37	25	0100101	%	Sinal de percentagem
38	26	0100110	&	"e" comercial
39	27	0100111	'	Apóstrofo
40	28	0101000	(Abre parênteses
41	28	0101001)	Fecha parênteses
42	2A	0101010	*	Asterisco
43	2B	0101011	+	Sinal mais
44	2C	0101100	,	Vírgula
45	2D	0101101	-	Hifen
46	2E	0101110	.	Ponto final
47	2F	0101111	/	Barra
48	30	0110000	0	
49	31	0110001	1	
50	32	0110010	2	
51	33	0110011	3	
52	34	0110100	4	
53	35	0110101	5	
54	36	0110110	6	
55	37	0110111	7	
56	38	0111000	8	
57	39	0111001	9	
58	3A	0111010	:	Dois pontos
59	3B	0111011	;	Ponto-e-vírgula
60	3C	0111100	<	Menor do que
61	3D	0111101	=	Igual a
62	3E	0111110	>	Maior do que

(continua)

Hidden page

Hidden page

Hidden page

Hidden page

Exemplo na Tabela B.2. Observe que os valores dos pesos são mostrados em potências de 2 e os valores correspondentes em decimal entre parênteses. O valor de um dígito específico é igual ao dígito vezes o peso de sua posição.

TABELA B.2 Pesos do sistema binário

Posição	Quinta	Quarta	Terceira	Segunda	Primeira
Peso	2^4 (16)	2^3 (8)	2^2 (4)	2^1 (2)	2^0 (1)

Para calcular o valor de um número, multiplicamos cada dígito pelo peso de sua posição, obtendo valores parciais para cada posição. Em seguida, somamos os resultados. A Figura B.3 demonstra a conversão do número binário 1101 para decimal. Como você pode ver, 1101 é o número binário equivalente ao número decimal 13.

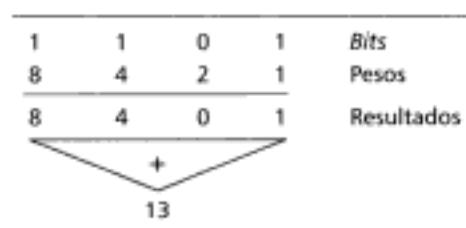


Figura B.3 Exemplo de um número binário.

Números Octais

O sistema de numeração octal é utilizado pelos programadores para representar os números binários numa forma compacta. Esse sistema também é conhecido como sistema *base 8*. A palavra octal deriva da palavra grega *octa* e significa 8. A utilidade desse sistema baseia-se no fato de que 8 é o resultado de uma potência de 2 (2^3) e assim pode ser usado para modelar conceitos binários. O sistema octal usa oito símbolos para representar os valores quantitativos: 0, 1, 2, 3, 4, 5, 6 e 7.

Os números octais usam oito símbolos: 0, 1, 2, 3, 4, 5, 6 e 7.

Pesos e Valores

O sistema octal também é um sistema com ponderação. Cada dígito possui um peso baseado na sua posição no número. O peso em octal é o valor da base (8) elevado à posição do dígito no número, como mostra a Tabela B.3. Observe que os valores dos pesos são mostrados em potências de 8 e os valores correspondentes em decimal entre parênteses. O valor de um dígito específico é igual ao dígito vezes o peso de sua posição. Por exemplo, um número 4 na terceira posição possui o valor equivalente decimal 4×64 ou 256.

TABELA B.3 Pesos do sistema octal

Posição	Quinta	Quarta	Terceira	Segunda	Primeira
Peso	8^4 (4096)	8^3 (512)	8^2 (64)	8^1 (8)	8^0 (1)

Para calcular o valor de um número, multiplicamos cada dígito pelo peso de sua posição, obtendo valores parciais para cada posição. Em seguida, somamos os resultados. A Figura B.4 demonstra a conversão do número octal 3471 para decimal. Como você pode ver, 3471 é o número octal equivalente ao número decimal 1849.

Hidden page

B.2 CONVERSÃO ENTRE BASES

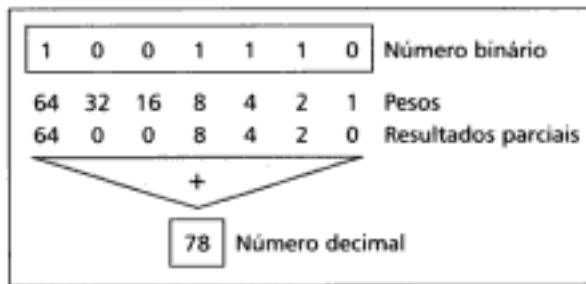
A não ser a compactação do que se está fazendo, não há nenhuma razão especial para preferirmos um sistema de numeração em detrimento de outros. São as aplicações que definem o tipo de sistema de numeração adotado. Um número dado em um sistema de numeração qualquer pode ser convertido no equivalente em outros sistemas. Por exemplo, um número binário pode ser convertido em um número decimal, e vice-versa, sem alterar o seu valor. A Tabela B.5 mostra como cada sistema representa os números decimais de 0 a 15. Como você pode ver, o decimal 13 é equivalente ao binário 1101, o qual representa o octal 15 e o hexadecimal D.

TABELA B.5 Comparação entre quatro sistemas de numeração

Decimal	Binário	Octal	Hexadecimal
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Convertendo de Outros Sistemas para Decimal

Como vimos nas discussões acima os números binário, octal e hexadecimal podem ser transformados facilmente nos seus equivalentes decimais usando os pesos dos dígitos. A Figura B.6 mostra o valor decimal 78 representado em cada um dos três sistemas descritos acima.



a. Binário para decimal

Figura B.6 Mudança de outros sistemas decimal (continua).

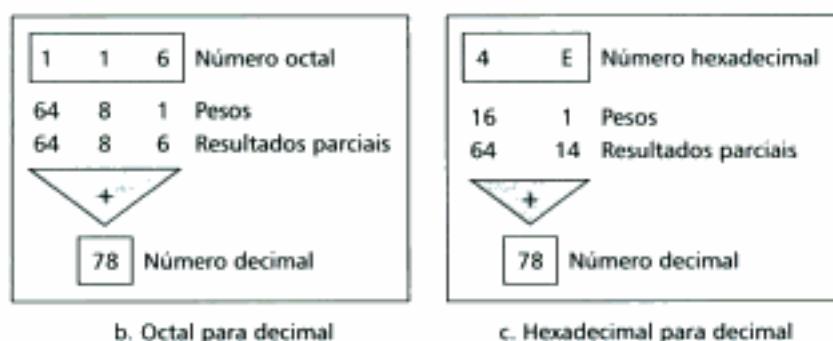


Figura B.6 Mudança de outros sistemas para decimal (continuação).

Convertendo de Decimal para Outros Sistemas

Uma simples divisão com resto fornece um modo bastante conveniente de converter um número decimal para o seu equivalente em binário, octal, hexadecimal ou outro sistema de numeração qualquer (veja a Figura B.7).

Para converter um número de decimal para binário, divida o número a ser convertido por 2 e escreva abaixo o resto resultante da divisão (1 ou 0). Esse resto da divisão será o dígito menos significativo. Em seguida, divida o quociente da primeira divisão por 2 e escreva abaixo o novo resto na segunda posição. Repita esse processo até que o quociente torne-se menor que o valor da base (2).

Na Figura B.7, convertemos o número decimal 78 para seu equivalente binário. Para verificar a validade desse método, vamos converter 1001110 para decimal, usando os pesos de cada posição.

$$2^6 + 2^3 + 2^2 + 2^1 \rightarrow 64 + 8 + 4 + 2 \rightarrow 78$$

Para converter um número de decimal para octal, o procedimento é o mesmo, mas o divisor é 8 e não 2. Para converter de decimal para hexadecimal, o divisor é 16.

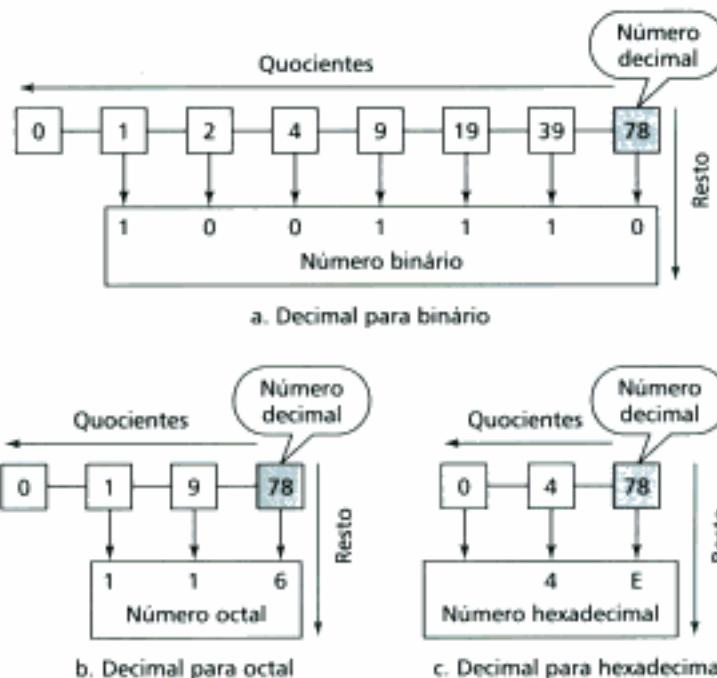


Figura B.7 Conversão de decimal para outros sistemas de numeração.

De Binário para Octal ou Hexadecimal

Para converter um número de binário para octal, primeiramente agrupamos em *tribits* os dígitos binários da direita para a esquerda. Então, convertemos cada *tribit* no seu equivalente em octal e escrevemos o resultado abaixo do *tribit*. Juntando todos os equivalentes dos *tribits*, colocados na ordem (e não adicionados), formamos o número octal equivalente ao número binário original. Na Figura B.8, convertemos o binário 1001110.

Para converter um número de binário para hexadecimal seguimos o mesmo procedimento, mas agrupamos os dígitos, da direita para a esquerda, em *tetrabits*. Dessa vez, convertemos cada *tetrabit* para o seu equivalente hexadecimal (use a Tabela B.5). Na Figura B.8, convertemos o número binário 1001110 para hexadecimal.

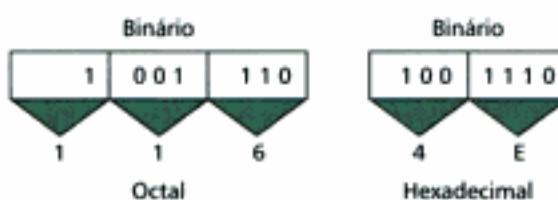


Figura B.8 Conversão de binário para octal ou hexadecimal.

De Octal para Hexadecimal ou Binário

Para converter de octal para binário invertemos o procedimento acima. Começando a partir do dígito menos significativo, convertemos cada dígito octal no seu equivalente binário de três dígitos. Na Figura B.9, convertemos o octal 116 para binário.

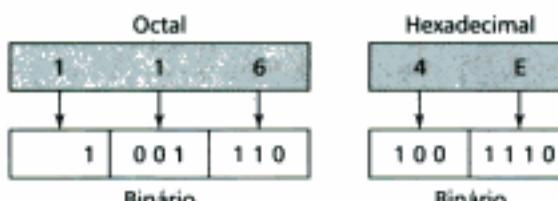


Figura B.9 Conversão de octal ou hexadecimal para binário.

Para converter de hexadecimal para binário, novamente começando a partir do dígito menos significativo, convertemos cada dígito hexadecimal no seu equivalente binário de três dígitos. Na Figura B.9, convertemos o octal 4E para binário.

O Modelo OSI

O modelo de camadas que dominou a literatura de comunicação de dados e redes de computadores antes de 1990 foi o **Modelo de Referência "Open Systems Interconnection" (interconexão de sistemas abertos) (MR-OSI)**. Todos acreditavam que o modelo OSI seria padrão de fato para comunicação de dados, mas isto acabou não acontecendo. O modelo da Internet ganhou força e tornou-se a arquitetura comercial porque ele foi usado e testado exaustivamente. O modelo OSI nunca recebeu sequer uma implementação e deve ser tratado apenas como modelo de referência.

Neste apêndice, discutimos rapidamente as principais características do modelo OSI em comparação com o modelo da Internet.

C.1 O MODELO

Estabelecida em 1947, a **International Organization for Standardization (ISO)** é um corpo multinacional de pesquisadores/projetistas dedicado à criação de padrões internacionais. Um padrão ISO que cobre todos os aspectos relacionados às redes de comunicação é o modelo Open Systems Interconnection (OSI). Este modelo foi apresentado no final da década de 70. Um **sistema aberto (open system)** é um conjunto de protocolos que permite a dois sistemas quaisquer se comunicarem não importando as arquiteturas em que eles estão baseados. O propósito do modelo OSI é mostrar como implementar comunicação entre dois sistemas sem requerer modificações lógicas tanto do ponto de vista de *hardware* quanto de *software*. O modelo OSI não é um protocolo; ele é um modelo que auxilia na compreensão e projeto de novas arquiteturas de rede que sejam ao mesmo tempo flexíveis, robustas e interoperáveis. A Figura C.1 apresenta o modelo de camadas OSI.

C.2 CAMADAS DO MODELO OSI

Nesta seção descreveremos brevemente as funções de cada camada do modelo OSI.

As Quatro Primeiras Camadas

As quatro primeiras camadas do modelo OSI (física, enlace, rede e transporte) e as camadas correspondentes no modelo da Internet são praticamente as mesmas. Por isso, não teceremos nenhum



Figura C.1 Modelo OSI.

comentário a respeito delas, pois o leitor tem ao longo do livro informações bastante detalhadas sobre elas.

ISO é a organização de padronização. **OSI** é o modelo.

Camada de Sessão

A **camada de sessão** é a camada *controladora de diálogo*. Ela estabelece, mantém e sincroniza a interação entre sistemas de comunicação.

As tarefas específicas da camada de sessão incluem:

- **Controle de diálogo.** A camada de sessão permite que dois sistemas iniciem um diálogo. Ela possibilita a comunicação entre dois processos no modo *half ou full-duplex*. Por exemplo, o diálogo entre um terminal burro com um *mainframe* acontece, na maioria das vezes, em modo *half-duplex*.
- **Sincronização.** A camada de sessão permite a um processo adicionar pontos de verificação (*checkpoints*) e de sincronização (*synchronization points*) em uma cadeia de dados. Por exemplo, se um sistema está transmitindo um arquivo de 2000 páginas, ele as divide inserindo *checkpoints* após cada conjunto de 100 páginas para assegurar que cada conjunto de 100 páginas é recebido e confirmado independentemente. Nesse caso, se ocorrer um *crash* durante a transmissão da página 523, as únicas páginas que devem ser retransmitidas são as páginas de 501 a 523. As páginas anteriores à 501 receberam confirmação e não precisam ser reenviadas. A Figura C.2 ilustra o relacionamento da camada de sessão com as camadas de transporte e de apresentação.

Camada de Apresentação

A **camada de apresentação** é responsável pela sintaxe e semântica da informação trocada entre dois sistemas. A Figura C.3 mostra o relacionamento entre a camada de apresentação e as camadas de aplicação e sessão.

As tarefas específicas da camada de apresentação incluem:

- **Tradução.** Os processos (programas executáveis) nos dois sistemas são usualmente a informação trocada na forma de caracteres *strings*, números e assim por diante. A informação deve ser convertida para uma cadeia de *bits* antes de ser transmitida. Devido ao fato de que diferentes tipos de computadores usam sistemas de codificação diferentes, a camada de apresentação é responsável pela interoperabilidade entre os diferentes métodos de codificação. A

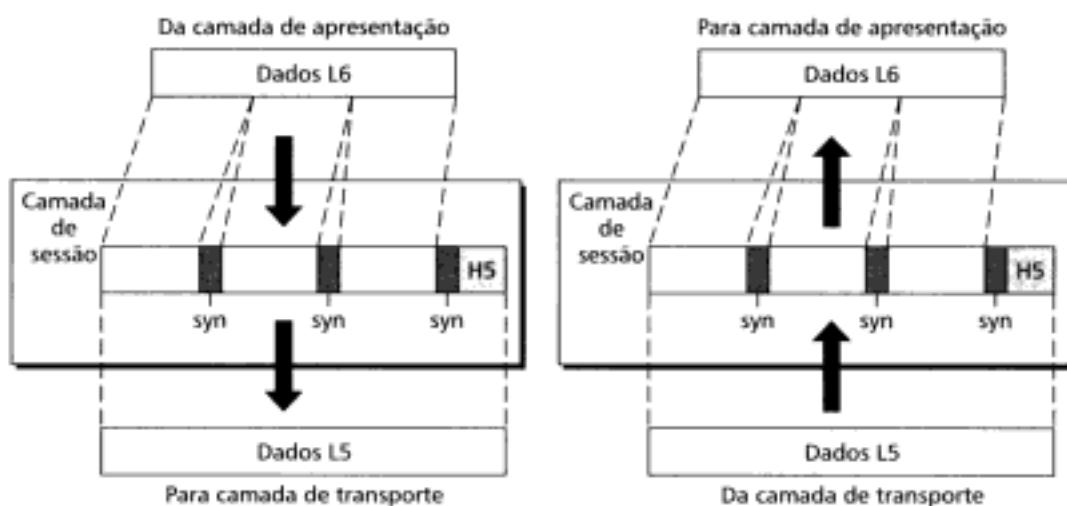


Figura C.2 Camada de sessão.

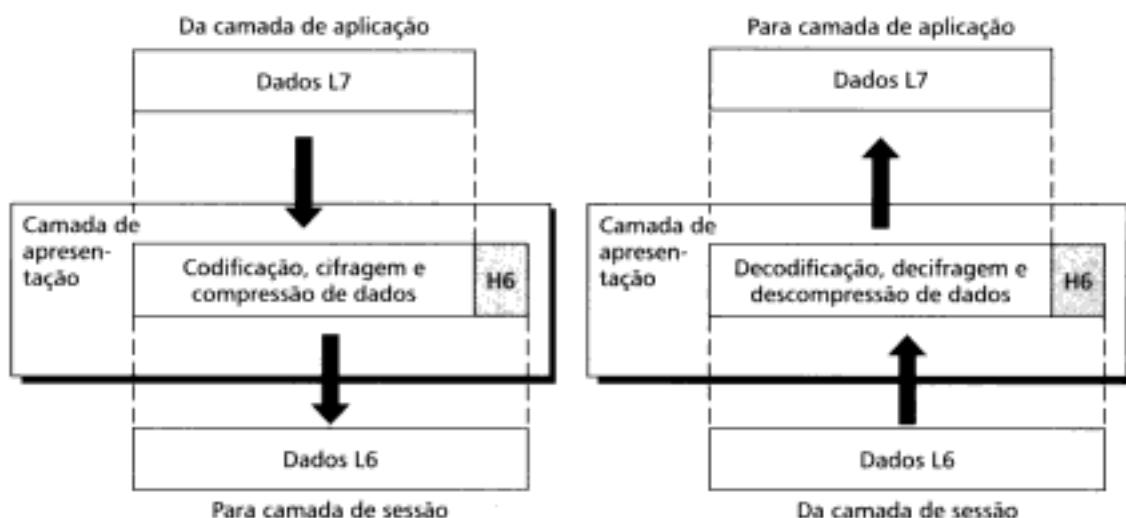


Figura C.3 Camada de apresentação.

camada de apresentação no transmissor converte a informação expressa no seu formato próprio numa informação em um formato comum (universal). A camada de apresentação no receptor converte a informação nesse formato universal para o formato adequado ao receptor.

- **Criptografia.** Para que seja possível o transporte de informação sensível, um sistema deve ser capaz de assegurar privacidade à comunicação. Criptografia possibilita ao transmissor converter a informação original para outra forma codificada e enviar a mensagem codificada resultante através da rede. Quando a mensagem chegar ao receptor ela deve passar por um processo de decifragem de volta para o formato original.
- **Compressão.** A compressão de dados reduz a quantidade de *bits* contida na informação. A compressão de dados é particularmente interessante nas transmissões de multimídia, tal como texto, áudio e vídeo.

Camada de Aplicação

A **camada de aplicação** permite que os usuários (seja uma pessoa ou um *software*) acessem a rede. Ela fornece interfaces aos usuários e suporte para serviços como *e-mail*, acesso e transferência remota de arquivos, compartilhamento e gerenciamento de bancos de dados e outros tipos de serviços de informação distribuída.

Hidden page

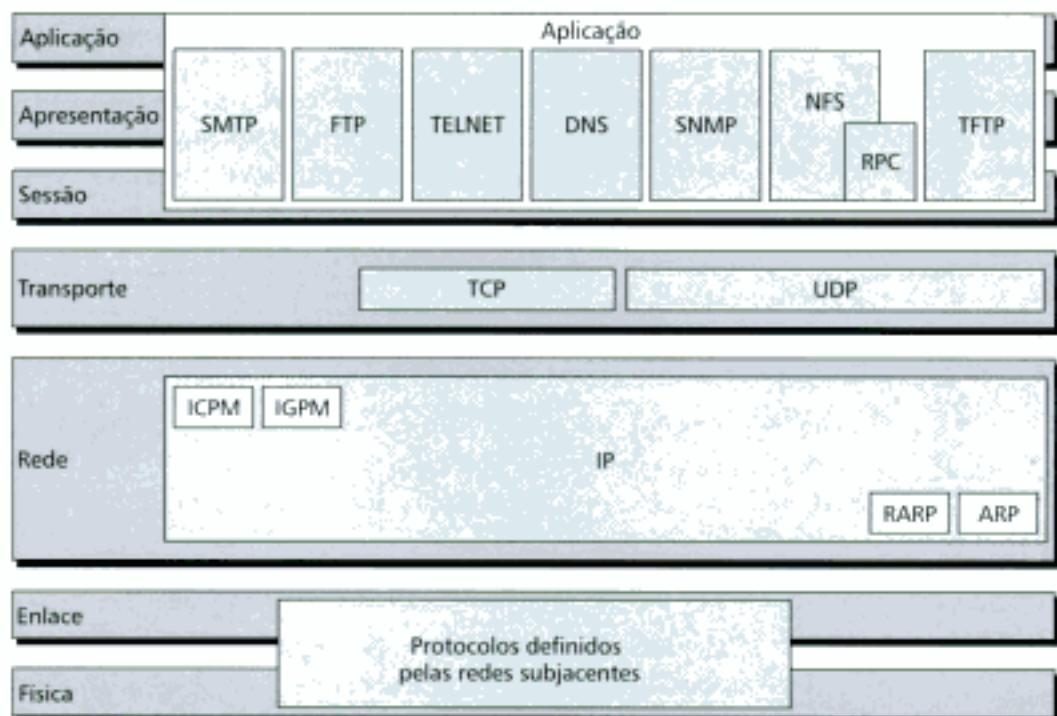


Figura C.5 A Internet e o modelo OSI.

Apêndice D

Código 8B/6T

Este apêndice é uma tabulação dos pares de códigos 8B/6T. Os 8 bits de dados são mostrados no formato hexadecimal. O código 8B/6T é mostrado com a notação + (sinal positivo), - (sinal negativo) e/ou 0 (sem sinal). Como a tabela é muito grande, mostramos a primeira metade do código na Tabela D.1 e a segunda metade na Tabela D.2.

TABELA D.1

Dados	Código	Dados	Código	Dados	Código	Dados	Código
00	-+00-+	20	-++-00	40	-00+0+	60	0++0-0
01	0-+-+0	21	+00+-	41	0-00++	61	+0+-00
02	0-+0-+	22	-+0-++	42	0-0+0+	62	+0+0-0
03	0-++0-	23	+--0++	43	0-0++0	63	+0+00-
04	-+0+0-	24	+--0+0	44	-00++0	64	0++00-
05	+0- -+0	25	-+0+00	45	00-0++	65	++0-00
06	+0-0-+	26	+00-00	46	00-+0+	66	++00-0
07	+0-+0-	27	-+ +--+	47	00-++0	67	++000-
08	-+00+-	28	0++-0-	48	00+000	68	0++-+-
09	0-++-0	29	+0+0--	49	++-000	69	+0++--
0A	0-+0+-	2A	+0+-0-	4A	+-+000	6A	+0+-+-
0B	0-+ -0+	2B	+0+- -0	4B	-++000	6B	+0+- -+
0C	-+0-0+	2C	0++--0	4C	0-+000	6C	0++- -+
0D	+0-+ -0	2D	+00- -	4D	+0-000	6D	++0+- -
0E	+0-0+-	2E	+ +0-0-	4E	0-+000	6E	++0- - -
0F	+0- -0+	2F	+ +0- -0	4F	-0+000	6F	++0- - -
10	0- - -+0+	30	+ -00-+	50	+--+0+	70	000++-
11	-0-0++	31	0+--+0	51	-+-0++	71	000+-+

(continua)

TABELA D.1 (continuação)

Dados	Código	Dados	Código	Dados	Código	Dados	Código
12	-0-+0+	32	0+-0-+	52	-++0+	72	000-++
13	-0-++0	33	0+-+0-	53	-+-++0	73	000+00
14	0--++0	34	+-0+0-	54	+- -+0+	74	000+0-
15	--00++	35	-0+-+0	55	--+0++	75	000+-0
16	--0+0+	36	-0+0-+	56	--++0+	76	000-0+
17	--0++0	37	-0++0-	57	--++0+	77	000-+0
18	-+0-+0	38	+-00+-	58	- -0+++	78	+++- -0
19	+ -0-+0	39	0+-+ -0	59	-0-+++	79	+++-0-
1A	-++-+0	3A	0+-0+-	5A	0-- +++	7A	++ +0--
1B	+00-+0	3B	0+- -0+	5B	0- -0++	7B	0++0--
1C	+00+-0	3C	+-0-0+	5C	+- -0++	7C	-00-++
1D	-+ + + -0	3D	-0++-0	5D	-000++	7D	-00+00
1E	+ -0+-0	3E	-0+0+-	5E	0+++- -	7E	+-+- + +
1F	-+0+-0	3F	-0+-0+	5F	0++-00	7F	+- -+00

TABELA D.2

Dados	Código	Dados	Código	Dados	Código	Dados	Código
80	-00++-	A0	-+ +0-0	C0	-+0+-+	E0	-+ +0-+
81	0-0-++	A1	+ -+ -00	C1	0-+ + +	E1	+ -+ -+0
82	0-0+-+	A2	+ +0-0	C2	0-+ + +	E2	+ -+0-+
83	0-0++-	A3	+ -+00-	C3	0-+ + +	E3	+ -+ +0-
84	-00++-	A4	-+ +00-	C4	-+0 + +	E4	-+ + +0-
85	00- -++	A5	+ + -00	C5	+0- - +	E5	+ + - -+0
86	00- + -	A6	+ + -0-0	C6	+0- + -	E6	+ + -0-+
87	00- + +	A7	+ + -00-	C7	+0- + +	E7	+ + -+0-
88	-000+0	A8	-+ + - +	C8	-+000+0	E8	-+ +0 + -
89	0-0+00	A9	+ -+ + -	C9	0-+ +00	E9	+ -+ + -0
8A	0-00+0	AA	+ -+ + -	CA	0-+0+0	EA	+ -+0 + -
8B	0-000+	AB	+ -+ - +	CB	0-+00+	EB	+ -+ -0 +
8C	-0000+	AC	-+ + - -	CC	-+000+	EC	-+ + -0 +
8D	00- +00	AD	+ + - + -	CD	+0- +00	ED	+ + - + -0
8E	00-0+0	AE	+ + - + -	CE	+0-0+0	EE	+ + -0 + -
8F	00-00+	AF	+ + - - +	CF	+0-00+	EF	+ + - -0 +
90	+ - - + -	B0	+000-0	D0	+ -0 + -	F0	+000 + -
91	-+ - - +	B1	0+0-00	D1	0+ - - +	F1	0+0- +0
92	-+ - + -	B2	0+00-0	D2	0+ - + -	F2	0+00- +
93	-+ - + + -	B3	0+000-	D3	0+ - + + -	F3	0+0 +0 -
94	-+ - + + -	B4	0+0000-	D4	0+ -0 + + -	F4	0+00 +0 -
95	- - + - + +	B5	00+ -00	D5	-0+ - + +	F5	00+ +0 +0
96	- - + - + +	B6	00+0-0	D6	-0+ + - +	F6	00+0 +0 +
97	- - + - + +	B7	00+00-	D7	-0+ + + -	F7	00+ +0 +0 -
98	+ - -0 +0	B8	+00- + -	D8	+ -00 +0	F8	+000 + - +

(continua)

Hidden page

Apêndice

E

Cálculo do Checksum

Este apêndice mostra como calcular um *checksum* tanto nas notações binária e hexadecimal.

E.1 NOTAÇÃO BINÁRIA

Usaremos a Figura E.1 para mostrar o cálculo do *checksum*.

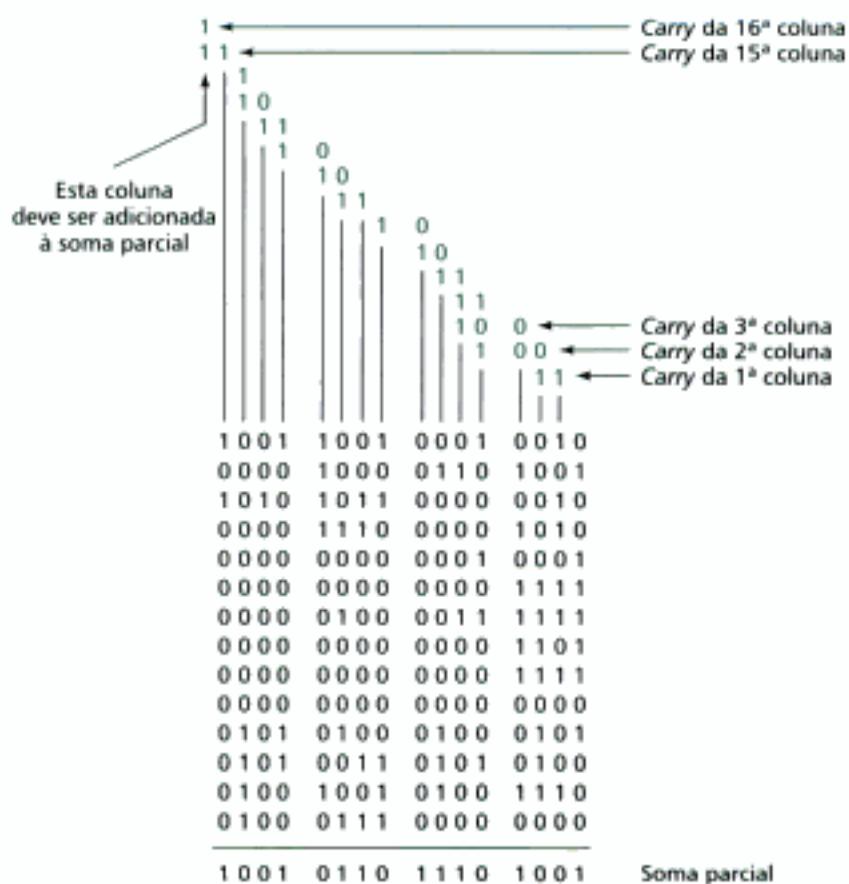


Figura E.1 Soma parcial na notação binária.

Hidden page

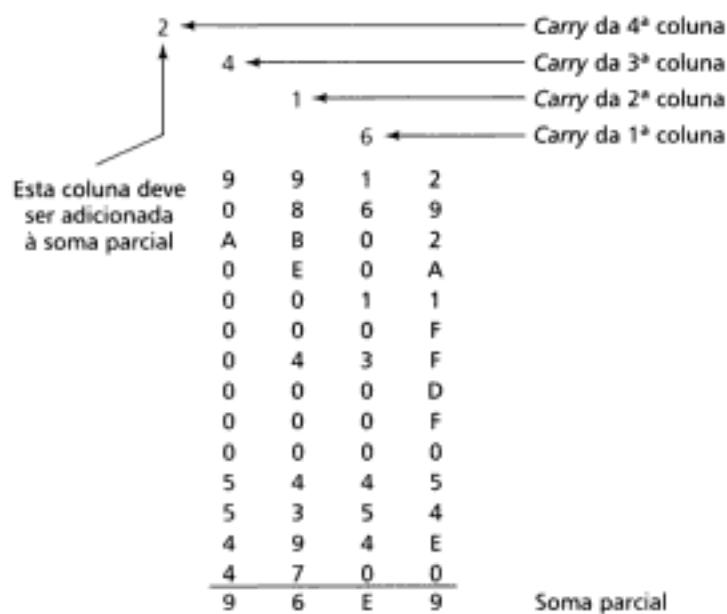


Figura E.3 Soma parcial na notação hexadecimal.

Sum

Se não houver *carry* da última coluna, a soma parcial é o próprio *sum*. Entretanto, se houver colunas extras (neste exemplo há uma coluna extra com duas linhas) os *bits* contidos nessa coluna devem ser adicionados à soma parcial para obter o *sum*. A Figura E.2 mostra este cálculo. Assim, obtemos o *sum*.

9	6	E	9	Soma parcial
			2	Carry da última coluna
9	6	E	B	Sum
6	9	1	4	Checksum
0 1 1 0	1 0 0 1	0 0 0 1	0 1 0 0	Checksum (binário)

Figura E.4 Sum e checksum na notação hexadecimal.

Checksum

Após o cálculo da soma (*sum*), complementamos cada dígito hexadecimal e obtemos o *checksum*. Observe que, quando calculamos o complemento, subtraímos cada dígito hexadecimal de 15 para obter o complemento (o equivalente ao complemento de um em hexadecimal). A figura também mostra a representação do *checksum* em binário.

Estrutura de um Roteador

Discutimos roteamento no Capítulo 19 e algoritmos de roteamento no Capítulo 21. Nesses dois capítulos representamos um roteador como uma caixa preta que recebe pacotes em uma de suas portas de entrada (interfaces), usa a tabela de roteamento para determinar a porta dos pacotes de partida e transmite os pacotes através dessa porta. Neste apêndice, vamos abrir a caixa preta e olhar dentro dela. Entretanto, nossa discussão não será tão detalhada ao ponto de cairmos no nível da eletrônica. Aos interessados existem livros inteiros sobre roteadores. Vamos dar apenas uma visão geral ao leitor.

F.1 COMPONENTES

Podemos dizer que um roteador possui quatro componentes: **portas de entrada**, **portas de saída**, **processador de roteamento** e **circuitos de comutação**, como mostra a Figura F.1.

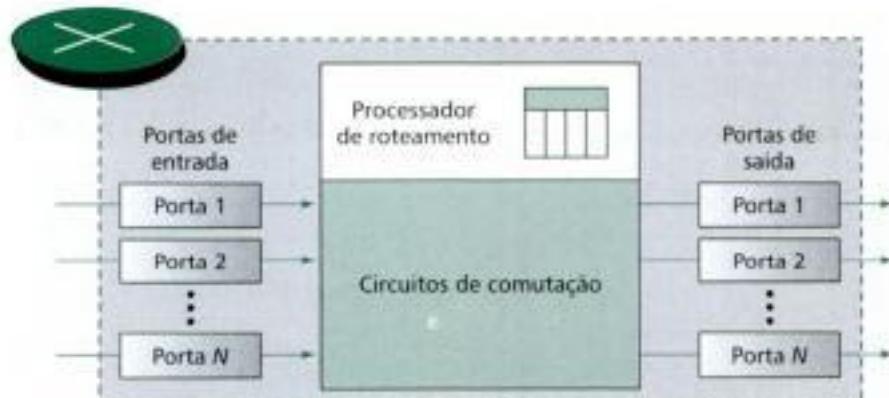


Figura F.1 Componentes de um roteador.

Portas de Entrada

Uma porta (interface) de entrada realiza as funções da camada física e de enlace do roteador. Os bits são reconstruídos a partir do sinal recebido. O pacote é desencapsulado do *frame*. Os erros são

detectados e corrigidos (caso existam). O pacote é lido pelo roteador na camada de rede. Além disso, para os processadores da camada física e de enlace, a porta de entrada possui *buffers* (filas) para manter os pacotes antes de dirigi-los aos circuitos de comutação. A Figura F.2 mostra um diagrama esquemático de uma porta de entrada.

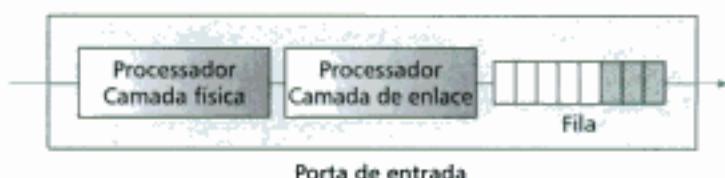


Figura F.2 Porta de entrada.

Portas de Saída

Uma porta (interface) de saída realiza as mesmas funções da porta de entrada, mas na ordem inversa. Primeiro, os pacotes de saída são colocados na fila. Então, o pacote é encapsulado em um *frame* e, finalmente, as funções da camada física são aplicadas ao *frame* para criar o sinal a ser transmitido no meio de transmissão. A Figura F.3 apresenta um diagrama esquemático de uma porta de saída.

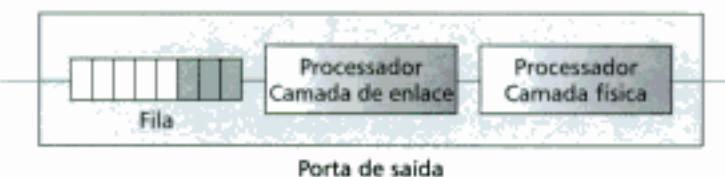


Figura F.3 Porta de saída.

Processador de Roteamento

O processador de roteamento realiza as funções da camada de rede. O endereço de destino é usado para determinar o endereço do próximo salto e, ao mesmo tempo, o número da porta de saída por onde o pacote será enviado. Esta atividade é denominada, com freqüência, *tabela de lookup*, porque o processador de roteamento procura rotas na tabela de roteamento. Nos roteadores de última geração, essa função está sendo deslocada para as portas de entrada para facilitar e agilizar o processo de expedição de pacotes.

Circuitos de Comutação

A tarefa mais difícil que um roteador enfrenta é mover eficientemente os pacotes da fila de entrada para a fila de saída. A velocidade de como isso é feito afeta o tamanho das filas de entrada/saída e, consequentemente, afeta toda a *performance* do roteador porque aumenta o atraso de entrega de pacotes. No passado, quando um roteador era de fato um computador dedicado, a memória do computador era utilizada como circuito de chaveamento. A porta de entrada armazenava o pacote na memória e a porta de saída retirava o pacote da memória para retransmiti-lo. Hoje, os roteadores são mecanismos especializados que usam uma variedade de circuitos de comutação. Faremos uma breve descrição desses circuitos nas próximas seções.

Comutador Matricial

O tipo mais simples de circuito de comutação é o comutador matricial discutido no Capítulo 8 e repetido na Figura F.4.

Hidden page

Hidden page

Hidden page

Arquitetura ATM Pura

Numa LAN ATM pura, um **comutador ATM** é utilizado para conectar as estações numa LAN, exatamente da mesma forma que as estações são conectadas em um *switch* Ethernet. A Figura G.2 apresenta a situação.



Figura G.2 LAN ATM pura.

Desse modo, as estações podem trocar dados em uma das duas taxas padronizadas pela tecnologia ATM (155 e 652 Mbps). Contudo, as estações devem utilizar um número **VPI (Virtual Path Identifier)** e outro número **VCI (Virtual Connection Identifier)** no lugar dos endereços de origem e de destino.

Arquitetura ATM Legada

Esta abordagem possui uma falha grave. O sistema precisa ser construído do zero; as LANs existentes em outras tecnologias não podem sofrer atualização para uma LAN ATM pura. A Figura G.3 ilustra esta arquitetura.

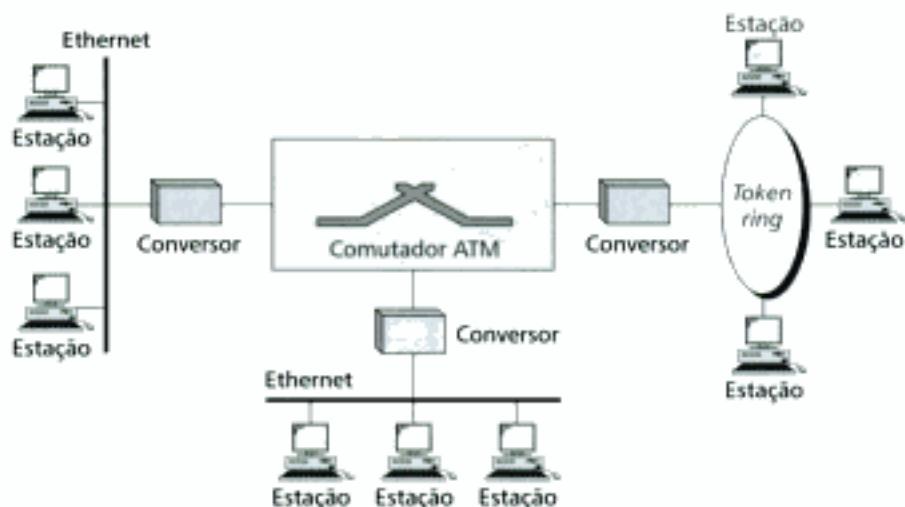


Figura G.3 LAN ATM herdada.

Assim, as estações na mesma LAN podem trocar dados em taxas e formatos das LANs padronizadas (Ethernet, Token Ring, etc.). Quando duas estações pertencentes a LANs diferentes necessitarem trocar dados, elas podem ir até um dispositivo conversor para modificar o formato do *frame*. A maior vantagem aqui é que as saídas de diversas LANs podem ser multiplexadas juntas para criar uma taxa de dados muito elevada em um comutador ATM. Há muitas questões que devem ser resolvidas antes da tecnologia LAN ATM ser colocada em prática.

Arquitetura Mista

Provavelmente, a melhor solução é o *mix* das duas arquiteturas anteriores. Isto significa manter as LANs existentes hoje e, ao mesmo tempo, permitir que novas estações sejam conectadas diretamente no comutador *switch*. A **arquitetura LAN mista** permite a graduação paulatina das LANs legadas para as LANs ATM, adicionando cada vez mais novas estações conectadas diretamente ao comutador. A Figura G.4 ilustra essa arquitetura.

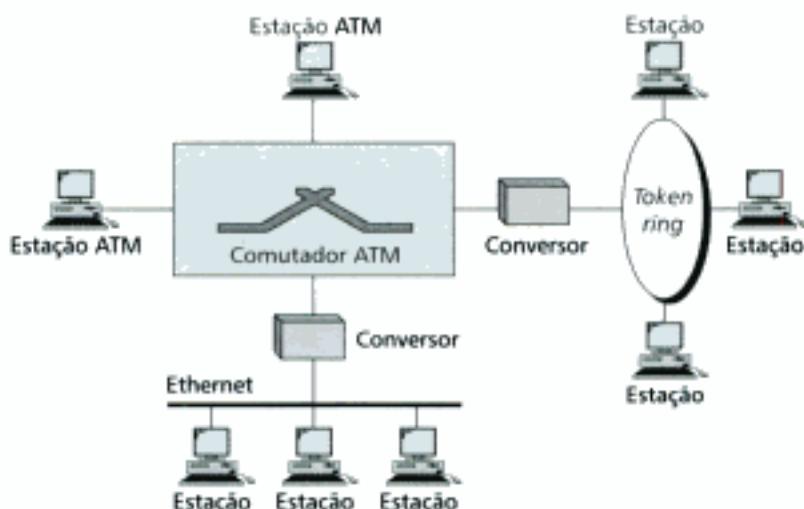


Figura G.4 Arquitetura LAN ATM mista.

Novamente, as estações numa LAN específica podem trocar dados usando o formato de texto de uma LAN particular. As estações diretamente conectadas ao comutador ATM podem utilizar um *frame* ATM para trocar dados. Porém, o problema é: uma estação numa LAN padrão pode se comunicar com a estação diretamente conectada? Véremos como o problema foi resolvido na próxima seção.

G.2 EMULAÇÃO DE LANS (LANE)

À primeira vista o uso da tecnologia ATM em LANs parece bem natural. Contudo, a similaridade é somente um ponto superficial; muitas outras questões precisam ser resolvidas, como resumimos abaixo:

- **Sem conexão versus orientado à conexão.** As LANs padrão, tal como a Ethernet, são protocolos **sem conexão**. Uma estação transmite pacotes de dados à outra estação sempre que os pacotes estiverem prontos. Não há nenhuma fase de **estabelecimento** ou **término de conexão**. Por outro lado, ATM é um **protocolo orientado à conexão**; a estação que desejar enviar células para outra estação deve primeiramente estabelecer uma conexão e, após todas as células serem enviadas, a conexão é finalizada.
- **Endereços físicos versus identificadores VCI.** A diferença entre os esquema de endereçamento é a primeira questão relacionada a LAN ATM. Um protocolo sem conexão, como o Ethernet, define a rota de um pacote através de **endereço de origem** e **endereço de destino**. Contudo, um protocolo orientado à conexão, tal como o ATM, define a rota de uma célula através de identificadores virtuais (VPIS e VCIS).
- **Processos multicasting e broadcasting.** Uma rede LAN padrão, como a Ethernet, pode enviar pacotes tanto em modo **multicast** quanto em **broadcast**; uma estação pode enviar pacotes a um grupo de estações ou a todas as estações de uma rede. Não há nenhuma for-

Hidden page

Broadcast/Unknown Server (BUS)

Multicasting e broadcasting requerem o uso de outro servidor denominado **Broadcast/Unknown Server (BUS)**. Se uma estação necessitar enviar um *frame* para um grupo de estações ou para cada estação, o *frame* é enviado primeiramente ao servidor BUS. Esse servidor possui uma conexão virtual permanente com cada estação. O servidor faz cópias do *frame* recebido e transmite uma cópia para cada grupo de estações ou para todas as estações, simulando um processo de *multicasting* ou *broadcasting*. O servidor também pode entregar um *frame unicast* enviando o *frame* para cada estação individualmente. Nesse caso, o endereço de destino é desconhecido (*unknown*). Às vezes, isso é mais eficiente que obter o identificador de conexão do LES.

G.4 ARQUITETURA MISTA CLIENTE-SERVIDOR

A Figura G.6 mostra clientes e servidores formando uma arquitetura LAN ATM mista. Na figura, três tipos de servidores estão conectados ao comutador ATM (de fato, eles podem ser parte do comutador). Na mesma figura mostramos dois tipos de clientes. As estações A e B, projetadas para enviar e receber comunicação LANE, são diretamente conectadas ao comutador ATM. As estações C, D, E, F, G e H nas LANs tradicionais estão conectadas ao comutador via um conversor. Tais conversores agem como clientes LEC e se comunicam em segurança com suas estações conectadas.

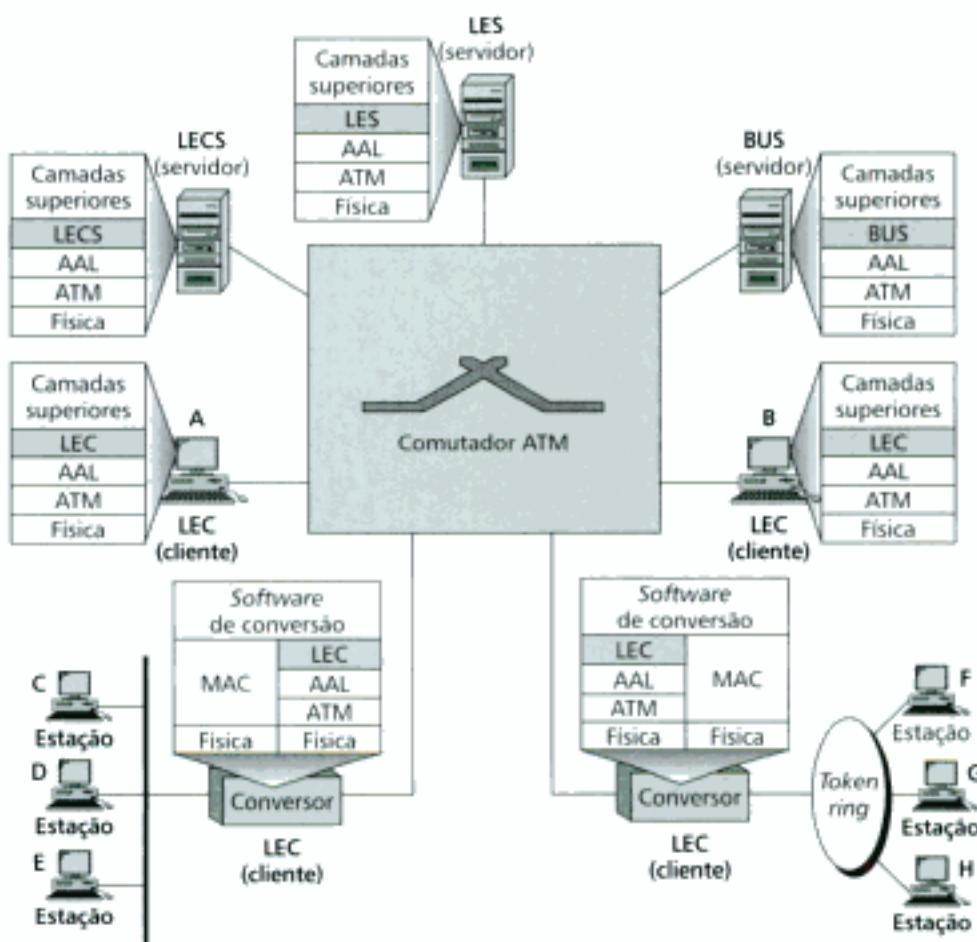


Figura G.6 Uma arquitetura LAN ATM usando LANE.

Hidden page

Programa H.1 (continuação)*Programa servidor UDP*

```

socklen_t clientAddrLen;
struct sockaddr serverAddr;
struck sockaddr clientAddr;
passiveSocket = socket (AF_INET, SOCK_DGRAM, 0);
memset (&serverAddr, 0, sizeof (serverAddr));
serverAddr.sin_family=AF_INET;
serverAddr.sin_port=htons (a-well-known-port);
serverAddr.sin_addr.s_addr=htonl (INADDR_ANY);
bind (passiveSocket, &serverAddr, sizeof (serverAddr));
clientAddrLen=sizeof (serverAddr);
memset (buf, 0, MAXBUF);
for ( ; ; )
{
    while (recvfrom (passiveSocket, buf, MAXBUF, 0,
        &clientAddr, &clientAddrLen)>0
    {
        PROCESS (.....);
        sendto (passiveSocket, buf , MAXBUF , 0 ,
            &clientAddr , clientAddrLen );
        memset (buf , 0 , MAXBUF );
    }
}
}
}

```

Programa Cliente UDP

O programa cliente é mostrado no programa H.2.

Programa H.2*Programa cliente UDP*

```

#include <sys/types.h>
#include <sys/socket.h>
#include <netdb.h>
#include <netinet/in.h>
#include <stdio.h>
#include <string.h>

#define MAXBUF 256

void main (void)
{
    char buf [ MAXBUF]
    int activeSocket ;

```

(continua)

Copyrighted material

Programa H.2 (continuação)*Programa cliente UDP*

```

socklen_t remoteAddrLen ;
struct sockaddr    remoteAddr ;
struct sockaddr    localAddr ;
struct hostent    *hptr ;
activeSocket= socket (AF_INET, SOCK_DGRAM, 0) ;
memset (&remoteAddr, 0 , sizeof (remoteAddr)) ;
remoteAddr.sin_family=AF_INET ;
remoteAddr.sin_port=htons (a-well-known-port) ;
hptr= gethostbyname (" a-domain-name") ;
memcpy ((char*) &remoteAddr.sin_addr.s_addr,
        hptr ->h_addr_list[0], hptr ->h_length ) ;

memset (buf , 0, MAXBUF);
remoteAddrLen=sizeof (remoteAddr) ;
while (get (buf) )
{
    sendto (activeSocket, buf , sizeof (remoteAddr) ,
            &remoteAddr , sizeof (remoteAddr) ) ;
    while (gets (buf) )
    {
        sendto (activeSocket, buf , sizeof (buf) , 0 ,
                &remoteAddr , sizeof (remoteAddr) ) ;
        memset (buf , 0, sizeof (buf) ) ;
        recvfrom (activeSocket, buf , MAXBUF, 0 ,
                  &remoteAddr , &remoteAddrLen ) ;
        printf ("%s\n", buf) ;
        memset (buf , 0 , sizeof (buf) ) ;
    }
    close (activeSocket) ;
}

```

H.2 PROGRAMAS CLIENTE-SERVIDOR TCP

Esta seção apresenta dois programas, um servidor e outro cliente. O servidor considerado é um servidor genérico. Temos definido um *PROCESS call* que é usado na construção da função servidora. Para simplificar, não mostramos nenhuma sentença de geração de mensagens de erros, tão necessárias nos programas reais.

Programa Servidor TCP

O programa H.3 mostra um programa servidor.

Hidden page

Hidden page

Apêndice

RFCs

Existem aproximadamente 2500 RFCs. Na Tabela 1.1 listamos em ordem alfabética de protocolos aquelas que estão relacionadas diretamente com o material deste texto. As RFCs principais de cada protocolo aparecem em negrito. Para a listagem completa, visite o site <http://www.faqs.org/rfcs>.

TABELA I.1 RFCs para cada protocolo

Protocolo	RFC
ARP e RARP	826, 903, 925, 1027, 1293, 1329, 1433
BGP	1092, 1105, 1163, 1265, 1266, 1267, 1364, 1392, 1403, 1565, 1654, 1655, 1665, 1745, 1997, 2238, 2439
BOOTP e DHCP	951, 1048, 1084, 1395, 1497, 1531, 1532, 1533, 1534, 1541, 1542, 2131, 2132
DHCP	Ver BOOTP e DHCP
DNS	799, 811, 819, 830, 881, 882, 883, 897, 920, 921, 1034, 1035, 1386, 1480, 1535, 1536, 1537, 1591, 1637, 1664, 1706, 1712, 1713, 1995, 2317
FTP	114, 133, 141, 163, 171, 172, 238, 242, 250, 256, 264, 269, 281, 291, 354, 385, 412, 414, 418, 430, 438, 448, 463, 468, 478, 486, 505, 506, 542, 553, 624, 630, 640, 691, 765, 913, 959, 1635, 2460, 2577
HTML	1866
HTTP	2068, 2109
ICMP	777, 792, 1016, 1018, 1256, 1788, 1885, 2521
IGMP	988, 1054, 1112, 2236
IP	760, 781, 791, 815, 950, 919, 922, 1025, 1063, 1141, 1190, 1191, 1624, 2113
IPv6	1365, 1550, 1678, 1680, 1682, 1683, 1686, 1688, 1726, 1752, 1826, 1883, 1884, 2133, 2147, 2492, 2553, 2590, 2675
MIME	Ver SNNIE, MIME, SMI
OSPF	1131, 1245, 1246, 1247, 1370, 1583, 1584, 1585, 1586, 1587, 2178, 2328, 2329, 2370
PIM	2362

(continua)

Hidden page

Portas UDP e TCP

A Tabela J.1 lista em ordem numérica as portas conhecidas TCP/UDP.

TABELA J.1 Portas TCP e UDP

Número da porta	UDP/TCP	Protocolo
7	TCP	ECHO
13	UDP/TCP	DAYTIME
19	UDP/TCP	CHARACTER GENERATOR
20	TCP	FTP-DATA
21	TCP	FTP-CONTROL
23	TCP	TELNET
25	TCP	SMTP
37	UDP/TCP	TIME
67	UDP	BOOTP-SERVER
68	UDP	BOOTP-CLIENT
69	UDP	TFTP
70	TCP	GOPHER
79	TCP	FINGER
80	TCP	HTTP
109	TCP	POP-2
110	TCP	POP-3
111	UDP/TCP	RPC
161	UDP	SNMP
162	UDP	SNMP-TRAP
179	TCP	BGP
520	UDP	RIP

A Tabela J.2 lista em ordem alfabética as portas conhecidas TCP/UDP.

TABELA J.2 Portas UDP e TCP

Protocolo	UDP/TCP	Número da porta
BGP	TCP	179
BOOTP-SERVER	UDP	67
BOOTP-CLIENT	UDP	68
CHARACTER GENERATOR	UDP/TCP	19
DAYTIME	UDP/TCP	13
ECHO	TCP	7
FINGER	TCP	19
FTP-DATA	TCP	20
GOPHER	TCP	70
HTTP	TCP	80
POP-2	TCP	109
POP-3	TCP	110
RIP	UDP	520
RPC	UDP/TCP	111
SMTP	TCP	25
SNMP	UDP	161
SNMP-TRAP	UDP	162
TELNET	TCP	23
TFTP	UDP	69
TIME	UDP/TCP	37



Endereços de Contato

Abaixo apresentamos uma lista de endereços de contato para várias organizações mencionadas no texto.

■ **FORUM ATM**

Presidio of San Francisco
PO. Box 29920 (correio)
572B Ruger Street
San Francisco, CA 94129-0920
Telefone: 415 561-6275
e-mail: info@atmforum.com
<http://www.atmforum.com>

■ **Federal Communications Commission (FCC)**

445 12th Street S.W.
Washington, DC 20554
Telefone: 1-888-225-5322
e-mail: fccinfo@fcc.gov
<http://www.fcc.gov>

■ **Institute of Electrical and Electronics Engineers (IEEE)**

Operations Center
445 Hoes Lane
Piscataway, NJ 08855-1331
Telefone: 732 981-0060
<http://www.ieee.gov>

■ **International Organization for Standardization (ISO)**

1, rue de Varembe
Case postable 56
CH-1211 Geneva 20
Switzerland
Telefone: 41 22 749 0111
e-mail: central@iso.ch
<http://www.iso.org>

■ International Telecommunication Union (ITU)

Place des Nations
CH-1211 Geneva 20
Switzerland
Telefone: 41 22 730 5852
e-mail: tsbmail@itu.int
<http://itu.int/ITU-T>

■ Internet Architecture Board (IAB)

e-mail: IAB@isi.edu
<http://www.iab.org>

■ Internet Corporation for Assigned Names and Numbers (ICANN)

4676 Admiralty Way, Suite 330
Marina del Rey, CA 90292-6601
Telefone: 310 823-9358
e-mail: icann@icann.org
<http://www.icann.org>

■ Internet Engineering Steering Group (IESG)

e-mail: iesg@ietf.org
<http://www.ietf.org/iesg.html>

■ Internet Engineering Task Force (IETF)

e-mail: ietf-infor@ietf.org
<http://www.ietf.org>

■ Internet Research Task Force (IRTF)

e-mail: irtf-chair@ietf.org
<http://www.irtf.org>

■ Internet Society (ISOC)

775 Weihle Avenue, Suite 102
Reston, VA 20190-5108
Telefone: 703 326-9880
e-mail: info@isoc.org
<http://www.isoc.org>

Acrônimos

4D-PAM5	4-Dimensional, 5-Level Pulse Amplitude Modulation	CFM	Cipher Feedback Mode
AAL	Application Adaptation Layer	CGI	Common Gateway Interface
ABM	Asynchronous Balanced Mode	CHAP	Challenge Handshake Authentication Protocol
ABR	Available Bit Rate	CIDR	Classless Inter-Domain Routing
ACK	Acknowledgment	CIR	Committed Information Rate
ACL	Asynchronous Connectionless Link	CLEC	Competitive Local Exchange Carrier
ADSL	Asymmetric Digital Subscriber Line	CMTS	Cable Modem Transmission System
AH	Authentication Header	CRC	Cyclic Redundancy Check
AM	Amplitude Modulation	CS	Convergence Sublayer
AMI	Alternate Mark Inversion	CSMA	Carrier Sense Multiple Access
AMPS	Advanced Mobile Phone System	CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
ANSI	American National Standards Institute	CSMA/CD	Carrier Sense Multiple Access with Collision Detection
API	Application Programming Interface	D-AMPS	Digital Amps
ARP	Address Resolution Protocol	DARPA	Defense Advanced Research Projects Agency
ARPA	Advanced Research Projects Agency	DB	Decibel
ARPANET	Advanced Research Projects Agency Network	DC	Direct Current
ARQ	Automatic Repeat Request	DCT	Discrete Cosine Transform
AS	Autonomous System Or Authentication Server	DDNS	Dynamic Domain Name System
ASCII	American Standard Code For Information Interchange	DDS	Digital Data Service
ASK	Amplitude Shift Keying	DE	Discard Eligibility
ATM	Asynchronous Transfer Mode	DEMUX	Demultiplexer
BECN	Backward Explicit Congestion Notification	DES	Data Encryption Standard
B-FRAME	Bidirectional Frame	DHCP	Dynamics Host Configuration Protocol
BGP	Border Gateway Protocol	Diffserv	Differentiated Services
BnZS	Bipolar n-Zero Substitution	DIFS	Distributed Interframe Space
BOOTP	Bootstrap Protocol	DLCI	Data Link Connection Identifier
BPS	Bits Per Second	DMT	Discrete Multi-Tone Technique
BSS	Basic Service Set	DNS	Domain Name System
CA	Certification Authority	DOCSIS	Data Over Cable System Interface Specification
CATV	Community Antenna TV	DS	Differentiated Services
CBC mode	Cipherblock Chaining mode	DSL	Digital Subscriber Line
CBR	Constant Bit Rate	DSLAM	Digital Subscriber Line Access Multiplex
CBT	Core-Based Tree	DSSS	Direct Sequence Spread Spectrum
CCITT	Consultive Committee for International Telegraphy and Telephony	DSU	Digital Service Unit
CCK	Complementary Code Keying	DVMRP	Distance Vector Multicast Routing Protocol
CDMA	Code Division Multiple Access	DWDM	Dense Wave Division Multiplexing

EBCDIC	Extended Binary Coded Decimal Interchange Code	IPv4	Internet Protocol, version 4
ECB	Eletronic Code Block	IPv6	Internet Protocol, version 6
EGP	Exterior Gateway Protocol	IR	Infrared
EIA	Electronics Industries Association	IRTF	Internet Research Task force
E-mail	Eletronic mail	IS-95	Interim Standard 95
ESP	Encapsulation Security Payload	ISO	International Organization Of Standardization
ESS	Estended Service Set	ISOC	Internet Society
FCC	Federal Communication	ITM-2000	Internet Mobile Communication for year 2000
FCS	Frame Check Sequence	ITU-T	International Telecommunications Union-Telecommunication Standardization Sector
FDDI	Fiber Distribuited Data Interface	IXC	Interexchange Carrier
FDM	Frequency Division Modulation	JPEG	Joint Photographic Experts Group
FDMA	Frequency Divison Multiple Access	KDC	Key Distribution Center
FE-CN	Foward Explicit Error Congestion Notification	LAN	Local Area Network
FHSS	Frequency Hopping Spread Spectrum	LANE	LAN Emulation
FIFO	First-In, First-Out	LCP	Link Control Protocol
FM	Frequency Modulation	LSA	Link State Advertisement
FQDN	Fully Qualified Domain Name	LSP	Link State Packet
FRAD	Frame Relay Assembler/ Dissembler	MA	Multiple Access
FSK	Frequency Shift Keying	MAC	address Physical address
FTP	File Transfer Protocol	MAN	Metropolitan Area Network
GMII	Gigabit Medium Independent	MAU	Medium Attachment Unit
GPS	Global Positioning System	MBONE	Multicast Backbone
GSM	Global System For Mobile Communication	MDI	Medium Dependent Interface
HDLC	High-level Data Link Control	MEO	Medium Earth Orbit
HDSL	High bit rate Digital Subscriber Line	MII	Medium Independent Interface
HFC network	Hybrid-Fiber-Coaxial network	MIME	Multipurpose Internet Mail Extension
HR-DSSS	High Rate Direct Sequence Spread Spectrum	MILT-3	Multiline Transmission, 3- level encoding
HTML	Hypertext Markup Language	MOSPF	Multicast Open Shortest Path First
HTTP	Hypertext Transfer Protocol	MPEG	Motion Picture Experts Group
Hz	Hertz	MSC	Mobile Switching Center
IAB	Internet Architecture Board	MTA	Mail Transfer Agent
IANA	Internet Assigned Numbers Authority	MTSO	Mobile Telephone Switching Office
ICANN	Internet Corporation For Assigned Names And Numbers	MTU	Maximum Transfer Unit
ICMP	Internet Control Message Protocol	NAK	Negative Acknowledgement
ICMPv6	Internet Control Message Protocol, version 6	NAT	Network Address Translation
IEEE	Institute Of Electrical And Eletronics Enginneer	NAV	Network Allocation Vector
IESG	Internet Engineering Steering Group	NCP	Network Control Protocol
IETF	Internet Engineering Task Force	NIC	Network Interface Card
I-frame	Intracoded frame	NNI	Network-to-Network Interface
IGMP	Internet Group Management Protocol	NRM	Normal Response Mode
IGP	Interior Gateway Protocol	NRZ	Nonreturn to Zero
ILEC	Incumbent Local Exchange Carrier	NRZ-I	Nonreturn to Zero-Invert
IMAP4	Internet Mail Access Protocol version 4	NRZ-L	Nonreturn to Zero-Level
INTERNIC	Internet Network Information Center	NSP	National Service Provider
Int Serv	Integrated Services	NVT	Network Virtual Terminal
IP	Internet Protocol	OC	Optical Carrier
IPCP	Internetwork Protocol Control Protocol	OSI	Open System Interconnection
Ipng	Internet Protocol, version 6	OSPF	Open Shortest Path First
IPSec	IP Security	PAM	Pulse Amplitude Modulation

Hidden page

Hidden page

Hidden page

- Algoritmo de Nagle** Algoritmo que tenta evitar a síndrome da janela boba no lado do transmissor. Nesse algoritmo, tanto a taxa de dados quanto a velocidade da rede são levadas em consideração.
- Algoritmo de procura** Regra para determinar o próximo salto.
- Algoritmo de *spanning tree*** Algoritmo que evita a formação de *loops* quando duas ou mais LANs são conectadas através de uma *bridge*.
- Algoritmo *leaky bucket*** Um algoritmo que modela o tráfego em rajadas.
- ALOHA** O método de acesso múltiplo aleatório original no qual uma estação podia enviar pacotes sempre que estivesse pronta para transmitir.
- Alternate Mark Inversion (AMI)** Codificação digital bipolar na qual os bits 1 são representados através de pulsos de tensão alternados (positivos e negativos).
- American National Standards Institute (ANSI)** Organização americana responsável pela padronização dentro dos Estados Unidos.
- American Standard Code for Information Interchange (ASCII)** Um código baseado em caracteres alfanuméricos desenvolvido pela ANSI e usado largamente na comunicação de dados.
- Amostragem** Processo de obtenção das amplitudes de um sinal em intervalos regulares.
- Amplitude de pico** O valor máximo de uma onda senoidal.
- Amplitude Shift Keying (ASK)** Um método de modulação no qual a amplitude do sinal da portadora é variado para representar 0 ou 1.
- Amplitude** A intensidade de um sinal, medido tipicamente em volts, ampères ou watts.
- Análise de Fourier** Uma técnica matemática usada para obter o espectro de frequência de um sinal aperiódico, caso seja fornecida a representação no domínio do tempo desse sinal.
- Analógico** Uma grandeza física variável continuamente.
- Ângulo crítico** Na refração, o valor do ângulo de incidência que produz um ângulo de refração de 90 graus.
- Ângulo de incidência** Na óptica, o ângulo formado entre raio de luz incidindo sobre uma superfície e uma linha paralela à superfície.
- Antena corneta** Captador direcional parecido com uma concha gigantesca. As transmissões são realizadas em multidifusão num tronco (guia de onda) e são defletidas numa série de feixes de ondas paralelas pela peça curvada na extremidade da antena.
- Antena de prato parabólico** Antena na forma de uma parábola utilizada nas comunicações terrestres via microondas.
- Antena de TV comunitária (CATV)** Uma rede de serviços a cabo que transmite sinais de vídeo em modo *broadcast* para localizações com pouca ou nenhuma recepção de sinais abertos.
- Antena omnidirecional** Antena que transmite ou recebe sinais em todas as direções.
- Applet** Um programa de computador para criar um documento ativo na Web. Usualmente é escrito em Java.
- Application Adaptation Interface (AAL)** Uma camada no protocolo ATM que quebra os dados do usuário em pequenos *payloads* de 48 bits.
- Application Programming Interface (API)** Um conjunto de declarações, definições e procedimentos seguido pelos programadores para escrever programas cliente-servidor.
- Área** Uma coleção de redes, hosts e roteadores, todos vinculados a uma sistema autônomo (AS).
- Arquitetura cliente-servidor** O modelo de interação entre dois programas aplicativos no qual um programa local (cliente) solicita um serviço de outro programa remoto (servidor).
- Arquitetura de camadas** Um modelo baseado em níveis ordenados.
- Arquivo imagem** No FTP, o formato padrão para transferência de arquivos binários. O arquivo é enviado como uma cadeia contínua de bits sem qualquer tipo de interpretação ou codificação.
- Árvore baseada na fonte** Árvore usada pelos protocolos de *multicasting* onde uma única árvore é formada para cada fonte que transmite a um grupo.
- Árvore de caminho mais curto** Tabela de roteamento montada com base no algoritmo Dijkstra.
- Árvore de menor custo** Uma característica do protocolo MOSPF na qual a árvore é baseada na escolha de uma métrica em vez do caminho mais curto.
- Assinatura digital** Um método de autenticação do transmissor de uma mensagem.
- Asymmetric Digital Subscriber Line (ADSL)** Uma tecnologia de comunicação na qual a taxa de dados em *download* é maior que a taxa em *upload*.
- Asynchronous Balanced Mode (ABM)** No protocolo HDLC, um modo de comunicação no qual todas as estações são iguais.
- Asynchronous Connectionless Link (ACL)** Um link entre um mestre e um escravo Bluetooth no qual um *payload* corrompido é retransmitido.
- Asynchronous Transfer Mode (ATM)** Um protocolo WAN caracterizado por altas taxas de transmissão e pacotes de dados pequenos (as células); é bastante apropriado para transmissão de texto, áudio e vídeo.

Hidden page

- Bluetooth** Uma tecnologia LAN sem fios (*wireless*) que conecta dispositivos de diferentes funcionalidades, tal como telefones e *notebooks*, em uma pequena sala.
- BnZS** Método de codificação que fornece sincronização para longas *strings* Os.
- Bootstrap Protocol (BOOTP)** Protocolo que fornece informação de configuração a partir de uma tabela (arquivo).
- Border Gateway Protocol (BGP)** Protocolo de roteamento inter-AS baseado no roteamento do vetor de caminhos.
- Bridge** Um dispositivo de rede operando nas duas primeiras camadas do modelo da Internet com capacidades de filtragem e encaminhamento de *frames*.
- Bridge remota** Dispositivo que conecta LANs e redes ponto a ponto; usado freqüentemente nas redes *backbone*.
- Bridge simples** Dispositivo de rede que conecta dois segmentos de rede, requerendo manutenção e atualização manual.
- Bridge transparente** Outro nome dado ao aprendizado da *bridge*.
- Broadcasting** Transmissão de uma mensagem a todos de uma rede.
- Browser** Um programa aplicativo que exibe um documento WWW. Um *browser* usa outros serviços da Internet para acessar um documento.
- Buffer de reprodução** Buffer que armazena dados até que eles estejam prontos para serem reproduzidos.
- Byte** Um grupo de oito bits.

C

- Cabeça de rede (*head end*)** Um centro de distribuição de TV a cabo.
- Cabeçalho** Informação de controle agregada a um pacote de dados; em um *e-mail*, a parte da mensagem que define o transmissor, o receptor, o assunto da mensagem e outras informações.
- Cabeçalho base** No protocolo IPv6, o cabeçalho principal do datagrama.
- Cabeçalho estendido** Cabeçalhos extras em um datagrama IPv6 que agrupa funcionalidade ao protocolo.
- Cabeçalho geral** Uma parte das mensagens pedido ou resposta do protocolo HTTP que fornece informação genérica sobre a mensagem.
- Cabeçalho pedido** Parte da mensagem pedido do protocolo HTTP que especifica a configuração e o formato dos documentos preferidos pelo cliente.
- Cabeçalho resposta** Parte da mensagem resposta do protocolo HTTP que especifica a configuração e as informações especiais sobre o pedido.
- Cable Modem Transmission System (CMTS)** Um dispositivo instalado dentro de um *hub* de distribuição que recebe dados da Internet e os repassa ao combinador.
- Cable modem** Uma tecnologia na qual a TV a cabo pode fornecer acesso à Internet.
- Cabo coaxial** Um meio de transmissão consistindo de um condutor central, um material isolante e um segundo condutor (uma malha).
- Cabo de fibra óptica** Um meio de transmissão com uma largura de banda elevada que transporta sinais na forma de pulsos de luz. Ele é construído de um núcleo cilíndrico fino de vidro ou plástico, encerrado em uma capa de proteção denominada *cladding*.
- Cabo par trançado** Meio de transmissão que consiste de dois condutores isolados e trançados um no outro.
- Cabo transceptor** No padrão Ethernet, o cabo que conecta a estação ao receptor. Também denominado um Attachment Unit Interface (AUI).
- Caching** Armazenamento de informação numa pequena porção de memória rápida.
- Camada ATM** Uma camada do modelo ATM que fornece serviços de roteamento, gerenciamento de tráfego, comutação e multiplexação.
- Camada de aplicação** A quinta camada do modelo da Internet; fornece acesso aos recursos da rede.
- Camada de apresentação** Sexta camada do modelo OSI responsável pela tradução, criptografia, autenticação e compressão de dados.
- Camada de enlace** A segunda camada no modelo da Internet. Ela é responsável pelas entregas *node-to-node* (entre os nós de uma rede).
- Camada de sessão** Quinta camada do modelo OSI, responsável por estabelecer, gerenciar e terminar as conexões lógicas entre dois usuários finais.
- Camada de transporte** A quarta camada do modelo da Internet e no modelo OSI; ela é responsável por agregar confiabilidade na comunicação entre processos finais.
- Camada path** Camada SONET responsável pelo deslocamento do sinal ótico da fonte ao destino.
- Camadas de suporte aos usuários** Camadas de aplicação, apresentação e sessão.
- Caminho de transmissão** Nas redes ATM, a conexão física entre dois comutadores.
- Caminho virtual (VP)** Numa rede ATM, uma conexão ou um conjunto de conexões entre dois *switches*.

Hidden page

- Cladding** Cobertura do núcleo de uma fibra óptica; material de proteção que envolve o núcleo da fibra.
- Classes de endereçamento** Um mecanismo de endereçamento no qual o espaço de endereços IP é dividido em 5 classes: A, B, C, D, E. Cada classe ocupa somente uma parte de todo o espaço de endereços.
- Classless InterDomain Routing (CIDR)** Uma técnica para reduzir a quantidade de entradas na tabela de roteamento quando é utilizado o endereço de *supernetting*.
- Cliente concorrente** Um cliente rodando ao mesmo tempo que outro cliente em um mesmo processo.
- Code Division Multiple Access (CDMA)** Um método de acesso múltiplo no qual um canal transporta todas as transmissões simultaneamente.
- Codificação bipolar** Um método de codificação digital no qual a amplitude 0 representa o binário 0 e as amplitudes positivas e/ou negativas representam a alternância de 1s.
- Codificação de blocos** Um método de codificação que garante a sincronização e detecção de erros.
- Codificação 2B1Q** Uma técnica de codificação de linha na qual cada pulso representa 2 bits.
- Codificação Huffman** Um método de compressão estatístico usando códigos de tamanho variável para codificar um conjunto de símbolos.
- Codificação Manchester** Método de codificação digital, a codificação Manchester usa uma inversão no meio de cada intervalo de sincronização tanto para a sincronização quanto a representação do bit.
- Codificação Manchester diferencial** Método de codificação digital que utiliza uma inversão no meio de cada intervalo de sincronização tanto para a sincronização quanto para a representação do bit 1.
- Codificação MLT-3 (Multiline Transmission, 3-level)** Esquema de codificação de linha caracterizado por 3 níveis de sinais e transições no início de 1 bit.
- Codificação 8B/10B** Uma técnica de codificação de blocos na qual 8 bits são codificados em um código de 10-bits.
- Codificação 8B/6T** Uma técnica de codificação na qual 8 bits são codificados em um código de 6-bits.
- Codificação polar** Método de codificação digital-analógico que utiliza dois níveis de tensão (positivo e negativo).
- Codificação preditiva** Na compressão de áudio, codificação das diferenças entre as amostras.
- Codificação 4B/5B** Uma técnica de codificação de bloco na qual 4 bits são codificados em outro código de 5 bits.
- Codificação unipolar** A codificação unipolar utiliza uma polaridade apenas. O sinal da polaridade pode ser atribuído a qualquer um dos dois estados binários, mas geralmente é deixado para o nível 1. Nesse caso, o outro estado (o nível 0) é representado por um zero de tensão.
- Código Hamming** Um método que adiciona bits redundantes à unidade de dados para detectar e corrigir erros.
- Colisão** O evento acontece quando dois transmissores tentam enviar dados ao mesmo tempo em um canal; os dados são destruídos.
- Committed Information Rate (CIR)** O tamanho da rajada é dividido no tempo.
- Common Gateway Interface (CGI)** Um padrão para comunicação entre servidores HTTP e programas executáveis. O CGI é utilizado na criação de documentos dinâmicos.
- Compartilhamento de árvore** Uma característica do roteamento *multicast* em que cada grupo do sistema compartilha a mesma árvore.
- Competitive Local Exchange Carrier (CLEC)** Uma companhia telefônica que não pode prover os principais serviços de telefonia; em vez disso, uma CLEC provê outros serviços como serviço de telefonia móvel.
- Complementary Code Keying (CCK)** Um método de codificação HR-DSSS que codifica quatro ou oito bits em um único, símbolo.
- Complemento de um** Representação de números binários em que o complemento do número é determinado através do complemento (inversão) de todos os bits do número.
- ***Componente DC** Veja *Corrente Contínua*.
- Compressão espacial** Compressão de imagens através da remoção das redundâncias.
- Compressão temporal** Método de compressão MPEG no qual os quadros redundantes são removidos.
- Comprimento de onda** Distância entre dois pontos de fases correspondentes numa forma de onda periódica.
- Comunicação entre nós da rede** Transferência de unidades de dados de um nó para o próximo.
- Comunicação entre processos finais** Entrega de pacotes do processo transmissor ao processo receptor.
- Comunicação wireless** Transmissão de dados usando meio não guiado.
- Comutação de circuitos virtuais** Uma técnica de comutação utilizada nas WANs comutadas.
- Comutação de circuitos** Uma tecnologia de comutação que estabelece uma conexão elétrica entre as estações usando um caminho dedicado.
- Comutação de pacotes** Transmissão de dados através de uma rede de comutação de pacotes.
- Comutação por divisão de espaço** Comutação na qual os caminhos são separados um dos outros espacialmente.
- Comutação por divisão do tempo** Técnica de comutação de circuitos onde a TDM é utilizada para obter a comutação.

- Comutador ATM** Dispositivo que fornece funções tanto de comutação quanto de multiplexação.
- Comutador matricial** Um comutador formado por conexões em linhas e colunas. Na interseção de cada linha e coluna (os pontos de cruzamento) acontece a conexão da entrada para a saída.
- Comutador multiestágios** Em arranjo de comutadores projetados para diminuir a quantidade de pontos de cruzamento (*crosspoints*).
- Conector BNC** Conector para cabo coaxial.
- Conector SC** Conector bastante de fibra óptica que utiliza os mecanismos de puxa/empurra.
- Conector ST** Tipo de conector muito utilizado em redes de fibra óptica.
- Conexão de controle** A conexão FTP usada para controle da informação (comandos e respostas).
- Conexão de dados** A conexão FTP usada para transferência de dados.
- Conexão local** O link que conecta um assinante a um central telefônica.
- Conexão não-persistente** Conexão na qual uma conexão TCP é estabelecida para cada solicitação/resposta.
- Conexão persistente** Conexão deixada aberta pelo servidor para que outras solicitações (pedidos) cheguem o envio de uma resposta.
- Conexão ponto a ponto** Link de transmissão dedicado entre dois dispositivos.
- Confiabilidade** Uma das características da QoS; dependente da transmissão.
- Configuração não balanceada** Em uma rede HDLC, a configuração na qual um dispositivo é o principal e o outro é o secundário.
- Confirmação (ACK)** Uma resposta enviada pelo receptor para indicar que ele recebeu e aceitou os dados enviados.
- Confirmação negativa (NAK)** Mensagem indicando a rejeição dos dados transmitidos.
- Congestion avoidance** Numa rede Frame Relay, um método que utiliza dois bits para notificar explicitamente a fonte e o destino sobre a ocorrência de congestionamento.
- Conjunto de protocolos** Pilha de protocolos definidos para um sistema de comunicação complexo.
- Conjunto de protocolos TCP/IP** Grupo de protocolos hierárquicos utilizados na Internet.
- Constelação** Uma representação gráfica da fase e da amplitude de combinações diferentes de bits em uma modulação digital-analógica.
- Consultative Committee for International Telegraphy and Telephony (CCITT)** Um grupo internacional de padronização atualmente conhecido como ITU-T.
- Contador de saltos** A quantidade de saltos para atingir uma determinada rede. Ele é a medida de distância em um algoritmo de roteamento baseado no vetor de distância.
- Contenção** Um método de acesso no qual dois ou mais dispositivos tentam transmitir ao mesmo tempo por um mesmo canal.
- Controle da conexão** A técnica usada pela camada de transporte para entregar segmentos.
- Controle de acesso** A determinação do controle do link através do protocolo de enlace.
- Controle de acesso ao meio (MAC)** A porção mais baixa da camada de enlace definida no projeto 802.2. Ele identifica o método e o controle de acesso ao meio em diferentes protocolos de rede LAN.
- Controle de congestionamento** Um método de gerenciamento de rede e do tráfego em uma internetworking para melhorar o throughput de uma rede.
- Controle de congestionamento em malha aberta** Políticas aplicadas na prevenção do congestionamento.
- Controle de congestionamento em malha fechada** Um método para reduzir o congestionamento (após ele ter ocorrido).
- Controle de erros** A detecção e o controle de erros em uma transmissão de dados.
- Controle de fluxo** Uma técnica para controlar o fluxo de frames (pacotes ou mensagens).
- Controle de tráfego** Método de modelamento e controle de tráfego em uma WAN.
- Controle do enlace** Uma das responsabilidades da camada de enlace: controle de fluxo e de erros.
- Controle do Link Lógico (LLC)** A porção mais alta da camada de enlace definida no Projeto IEEE 802.2.
- Convergência lenta** A velocidade e a habilidade de um grupo de dispositivos de internetworking que executa um protocolo de roteamento específico para concordar quanto à topologia de uma internetwork depois de uma troca nessa topologia.
- Conversão Analógico-Digital** A representação da informação analógica por um sinal digital.
- Core-Based Tree (CBT)** No multicasting, um protocolo de compartilhamento de grupo que usa os serviços de um roteador central como roteador raiz da árvore.
- Correção de erros por retransmissão** O processo de corrigir bits corrompidos através da retransmissão de dados.
- Correção direta de erros** Correção de erros feita pelo receptor.
- Correio eletrônico (*e-mail*)** Um método de envio de mensagens eletrônicas baseado nos endereços das caixas de correio eletrônico e não nos endereços dos hosts.
- Corrente Contínua (DC)** Sinal de frequência zero com uma amplitude constante.

Criptografia com chave secreta Método de segurança segundo o qual a chave de cifragem é a mesma que a chave de decifragem da mensagem. Tanto o transmissor quanto o receptor devem possuir as mesmas chaves.

Criptografia com chave simétrica Cifra na qual uma mesma chave é utilizada para cifrar e decifrar uma mensagem.

Criptografia de chave pública Método de cifragem baseado em um algoritmo de criptografia não reversível. O método usa dois tipos de chaves: a chave pública de domínio público e a chave privada (chave secreta), conhecida apenas pelo receptor.

Criptografia A ciência e a arte da conversão de mensagens para torná-las seguras e imunes a ataques.

Crosstalk O ruído em uma linha causado por sinais viajando ao longo de outra linha.

CSNET Uma rede patrocinada pela National Science Foundation, dedicada originalmente às universidades.

Cyclic Redundancy Check (CRC) Um método de correção de erros altamente apurado baseado na interpretação do padrão de bits como um polinômio.

D

Dados analógicos Informações que variam continuamente e não possuem uma quantidade limitada de valores.

Dados digitais Dados representados por valores discretos.

Dados em upstream Numa rede HFC, a banda de 5 a 42 MHz para transmissão de dados do assinante para a Internet.

Data Encryption Standard (DES) Método padrão de criptografia adotado pelo governo americano para uso não militar e não classificado.

Data Link Connection Identifier (DLCI) Um número que identifica um circuito virtual em uma rede Frame Relay.

Data Over Cable System Interface Specifications (DOCSIS) Um padrão para a transmissão de dados sobre uma rede HFC.

Datagrama É uma unidade de dados independente em uma rede de comutação de pacotes.

Datagrama IP A unidade de dados do protocolo IP.

Datagrama UDP O nome do pacote no protocolo UDP.

Decibel (dB) Uma medida da intensidade relativa do sinal em dois pontos diferentes.

Decifragem Nome do processo de recuperação da mensagem original a partir da informação cifrada.

Defense Advanced Research Projects Agency (DARPA) Uma organização governamental que, sob o nome ARPA, fundou a ARPANET e a Internet.

Demodulação O processo de separação da portadora do sinal de informação original (sinal modulante).

Demodulador Um dispositivo que realiza a demodulação.

Demultiplexador (DEMUX) Um dispositivo que separa os sinais multiplexados de volta nas componentes originais.

Dense Wave-Division Multiplexing (DWDM) Um método WDM que pode multiplexar uma quantidade muito grande de canais ópticos.

Descartar elegibilidade (DE) Um bit que define que um pacote pode ser descartado se houver congestionamento em uma rede.

Diagrama de transição de estados Diagrama que ilustra os estados de uma máquina de estados finitos.

Dibit Uma unidade de dados formada por dois bits.

Digest Veja *Síntese da mensagem*.

Digital AMPS (D-AMPS) Sistema de telefonia celular de segunda geração que é a versão digital do padrão AMPS.

Digital Data Service (DDS) Uma versão digital de uma linha dedicada com uma taxa de 64 kbps.

Digital Service Unit (DSU) Um dispositivo que permite a conexão do dispositivo de um usuário a uma linha digital.

Digital Subscriber Line (DSL) Linha digital do assinante (essa tecnologia usa as redes de telecomunicações existentes para realizar transmissão em alta velocidade de dados, voz e multimídia).

Direct-Sequence Spread Spectrum (DSSS) Um método de transmissão wireless no qual cada bit a ser enviado pelo transmissor é substituído por uma sequência de bits denominada chip code.

Discagem Processo utilizado nas ligações telefônicas.

Discagem por teclado Método de realização de uma chamada telefônica que utiliza um teclado para a discagem dos números.

Discrete Cosine Transform (DCT) Uma fase do processo JPEG na qual uma transformação modifica os 64 valores de uma matriz, mantendo os relacionamentos relativos entre pixels, para evidenciar as redundâncias.

Discrete Multitone Technique (DMT) Um método de modulação que combina elementos das técnicas QAM e FDM.

Dispositivo de conexão Um dispositivo que conecta computadores ou redes.

Distance Vector Multicast Routing Protocol (DVMRP) Um protocolo baseado no vetor de distância que controla o roteamento multicast em conjunto com o IGMP.

Distorção Qualquer modificação no sinal original provocado por ruído, atenuação ou outras influências.

Distributed Interframe Space (DIFS) Em uma rede wireless, o intervalo de tempo que uma estação espera antes de enviar um frame de controle.

Documento ativo Na WWW, um documento executado no site local usando Java.

Documento dinâmico Um documento da Web criado por um programa CGI rodando no site do servidor.

Documento estático Documento com conteúdo fixo criado e armazenado em um servidor WWW.

Domain Name System (DNS) Uma aplicação TCP/IP que converte nomes em endereços IP.

Domínio de colisão O tamanho do meio sujeito à colisão.

Domínio de reserva Um subdomínio do DNS que determina o nome do domínio a partir do endereço IP.

Domínio geográfico Um subdomínio DNS que usa dois caracteres identificadores do país como sufixo.

Domínio organizacional Um subdomínio em um sistema de nomes de domínio que usa sufixos genéricos.

Downlink Transmissão de um satélite até uma estação situada no solo.

Downloading Processo no qual um usuário baixa um arquivo de um site remoto.

Dynamic Domain Name System (DDNS) Um método para atualizar dinamicamente o arquivo DNS mestre.

Dynamic Host Configuration Protocol (DHCP) Uma extensão do protocolo BOOTP que atribui informação dinâmica de endereçamento IP aos clientes de uma rede.

E

Electronic Code Block (ECB) Um modo de operação DES ou 3DES em que uma mensagem longa é dividida em blocos menores de 64 bits para serem cifrados separadamente.

Electronics Industries Association (EIA) Uma organização que cuida dos aspectos relacionados à manufatura eletrônica nos Estados Unidos. Ela desenvolveu vários padrões para interfaces, dentre eles EIA-232, EIA-449 e EIA-530.

e-mail Veja *Correio eletrônico*.

Encapsulamento A técnica na qual uma unidade de dados (*payload*) de um protocolo é colocada dentro do campo de unidade de dados de outro protocolo.

Endereçamento sem classes Um mecanismo de endereçamento no qual o espaço de endereços IP não é dividido em classes.

Endereço anycast Um endereço que define um grupo de computadores cujos endereços têm a mesma estrutura inicial.

Endereço classe A Um endereço IPv4 cujo primeiro octeto está situado em 0 e 127 (em decimal).

Endereço classe B Um endereço IPv4 cujo primeiro octeto está situado em 128 e 191 (em decimal).

Endereço classe C Um endereço IPv4 cujo primeiro octeto está situado em 192 e 223 (em decimal).

Endereço classe D Um endereço IPv4 *multicast*.

Endereço classe E Um endereço IPv4 reservado à pesquisa experimental.

Endereço da sub-rede Endereço da rede de uma sub-rede.

Endereço de broadcast Um endereço que permite transmissão de uma mensagem a todos os hosts de uma rede.

Endereço de Internet Um endereço de 32 bits (IPv4) ou 128 bits (IPv6) usado para definir unicamente um host em uma rede TCP/IP.

Endereço de multicast Um endereço utilizado para a operação de *multicasting* (veja também *multicasting*).

Endereço de origem O endereço do transmissor da mensagem.

Endereço de porta No protocolo TCP/IP um número inteiro identificando um processo.

Endereço de serviço Veja *endereço de porta*.

Endereço de socket Estrutura formada por um endereço IP e um número de porta TCP.

Endereço físico Veja *Controle de acesso ao meio*.

Endereço local A parte de um endereço de e-mail que define o nome de um arquivo especial, denominado caixa de correio, onde todos os e-mails recebidos pelos usuários são armazenados pelo *user agent*.

Endereço lógico Um endereço definido na camada de rede.

Endereço unicast Endereço pertencente a um só destino.

Entrega fonte ao destino Transmissão de uma mensagem do transmissor original ao receptor desejado.

Entrega hop-to-hop Transmissão de frames de um nó para o próximo.

Erro isolado Tipo de erro na unidade de dados onde somente um bit é alterado por vez.

Escravo Estação sob o controle de um mestre em uma rede piconet.

Espaço de endereços O número total de endereços usado por um protocolo.

Espaço de nomes Todos os nomes atribuídos às máquinas dentro de uma internet.

Espaço de nomes do domínio Uma estrutura para organizar o espaço de nomes na qual os nomes do domínio são definidos em uma estrutura de árvore invertida com a raiz no topo.

- Espaço de nomes hierárquico** Um espaço de nomes obedecendo uma hierarquia de nomes centralizadas em servidores de nomes, todos remetidos a um servidor raiz.
- Espaço de nomes plano** Um método que mapeia um nome em um endereço IP sem a utilização de hierarquia de nomes.
- Espectro** Faixa de freqüências de um sinal.
- Espectro eletromagnético** A faixa de freqüências em que a energia eletromagnética está distribuída.
- Estabelecimento da conexão** Uma configuração preliminar necessária à conexão lógica entre duas entidades de uma rede.
- Estação móvel** Um *host* que pode se mover entre as redes.
- Estação principal** No método de acesso principal/secundário, uma estação que transmite comandos às estações secundárias.
- Estação secundária** No método de acesso *poll/select*, uma estação que envia uma resposta a um determinado comando de uma estação primária.
- Estado de autenticação** No protocolo PPP, um estado opcional que verifica a identidade do receptor.
- Estado de estabelecimento da conexão** No PPP, um estado no qual a comunicação é iniciada e as opções são negociadas.
- Estado de rede** Estágio PPP no qual pacotes de dados do usuário e pacotes de controle sejam transmitidos.
- Estado de terminação** Estado do protocolo PPP onde muitos pacotes são trocados entre as duas extremidades para ajustes e término da conexão.
- Estado ocioso** No PPP, um estado no qual o *link* está inativo.
- Estratégia de atraso de resposta** Uma técnica utilizada pelo IGMP para evitar tráfego desnecessário em uma LAN.
- Estratégia de persistência** Estratégia adotada pelo método CSMA em que uma estação transmite um *frame* toda vez que escutar o meio.
- Estratégia não-persistente** Método de acesso segundo o qual uma estação espera um tempo aleatório, após uma colisão, antes de reiniciar a transmissão de pacotes.
- Estratégia p-persistente** Uma estratégia CSMA persistente na qual uma estação transmite com probabilidade p se ela encontrar o meio ocioso.
- Estratégia 1-persistente** Uma estratégia de persistência CSMA na qual uma estação envia um *frame* imediatamente se a linha estiver ociosa.
- Ethernet FL** Padrão Ethernet usando fibra óptica.
- Ethernet par trançado** Rede Ethernet usando cabo par trançado 10Base-T.
- Ethernet** Um padrão de rede local que usa o método de acesso CSMA/CD. Veja *Projeto IEEE 802.3*.
- Extended Service Set (ESS)** Um serviço WLAN composto de duas ou mais BSSs cujos pontos de acesso (*access point*) foram definidos no padrão IEEE 802.11.
- Extranet** Uma rede privada que usa os serviços do protocolo TCP/IP para autorizar o acesso aos usuários externos.

F

- Fase** Posição relativa de um sinal no tempo.
- Fase de configuração** Fase onde uma origem e um destino de dados usam os respectivos endereços globais para auxiliá-los a preencher as entradas em tabela de modo a estabelecer uma conexão através de uma rede de comutação de circuitos virtuais.
- Fase de desconexão** Na comutação de circuitos virtuais, a fase na qual uma fonte e um destino informam ao comutador para apagar as suas entradas da tabela.
- Fast Ethernet** Veja *100Base-T*.
- Fator de reuso** Em uma rede de telefonia celular, quantidade de células que podem ser reutilizadas com conjuntos de freqüências diferentes.
- Federal Communication Commission (FCC)** Uma agência do governo americano que regulamenta as transmissões de rádio, TV e as telecomunicações.
- Fiber Distributed Data Interface (FDDI)** Uma LAN de alta velocidade (100 Mbps) definida pela ANSI, usando fibras ópticas, topologia em anel duplo e o método de acesso por passagem da permissão. Atualmente, uma rede FDDI é utilizada como uma MAN.
- Fibra multimodo índice degrau** Fibra óptica onde o nível de distorção do sinal provocado pelo movimento através do cabo é diminuído. Uma fibra com índice gradual é aquela onde o índice de refração varia gradualmente. A densidade é mais alta no centro do núcleo e vai diminuindo gradualmente até atingir um valor mínimo na interface núcleo/casca.
- Fibra multimodo índice gradual** Fibra óptica onde a densidade do núcleo permanece constante do centro até a borda da interface núcleo/casca. Na interface, ocorre uma mudança abrupta para uma densidade menor que altera o ângulo de movimento do feixe.
- Fibra óptica monomodo** Fibra óptica com um diâmetro extremamente pequeno que limita a propagação do feixe de luz praticamente à propagação horizontal.

Hidden page

Grafting (enxerto) Recomeço das mensagens de *multicast*.

Grupo Um sinal analógico formado por 12 canais de voz multiplexados.

Grupo jumbo Um sinal analógico criado a partir da multiplexação de seis grupos mestres.

Grupo mestre Sinal analógico formado pela multiplexação de 10 supergrupos.

H

H.323 Um padrão desenvolvido pelo ITU-T que permite interligar telefones da rede pública com computadores (denominados terminais no H.323) conectados à Internet.

Handoff Mudança para um novo canal de comunicação quando uma estação móvel (telefone celular) sai de uma célula e entra em outra.

Harmônicos Componentes de freqüência de um sinal digital, cada qual tendo amplitude, fase e freqüência diferentes.

Hertz (Hz) Unidade de medida de freqüência.

Hierarquia analógica Um sistema de telefonia na qual os sinais multiplexados são combinados em grupos sucessivamente maiores para melhorar a eficiência da transmissão.

High Bit Rate Digital Subscriber Line (HDSL) Um serviço similar à linha T-1 que pode atingir até 3,6 km.

High Level Data Link Control (HDLC) Um protocolo de enlace orientado a *bit* definido pela ISO. Ele é utilizado no protocolo X.25. Um subconjunto desse protocolo, denominado LAP é utilizado em outros protocolos. Ele também serve como base para muitos outros tipos de protocolos utilizados nas WANs.

High Rate Direct Sequence Spread Spectrum (HR-DSSS) Um método de geração de sinal similar ao DSSS, exceto que utiliza o método de codificação CCK.

Hipermídia Informação contendo texto, figuras, gráficos e áudio que são vinculados a outros documentos através de ponteiros.

Hipertexto Informação contendo texto vinculado a outros documentos através de ponteiros.

Homepage Uma unidade de hipertexto ou hipermídia disponível na Web. Página principal de um indivíduo ou organização.

Host Uma estação ou nó de uma rede.

Host específico nesta rede Endereço especial onde a *netid* é formada toda de zeros.

Host_file Um arquivo, usado quando a Internet ainda era pequena, para mapear nomes de *host* em endereços.

Host permanente Host conectado permanentemente a uma rede.

Host remoto Computador que um usuário deseja acessar a partir de outro computador (geralmente local).

Hub Dispositivo concentrador da topologia de rede em estrela.

Hub de distribuição Em uma rede HFC, um local onde os sinais são modulados e distribuídos.

HyperText Markup Language (HTML) A linguagem de computador que especifica os conteúdos e formatos de um documento da Web. Ela permite que sejam adicionados textos para incluir códigos que definem as fontes, layouts, gráficos e hipertexto.

HyperText Transfer Protocol (HTTP) Um protocolo de aplicação que permite manipular os documentos da Web.

I

Identificação de área Um campo de 32 bits que define a área dentro da qual o roteamento deve acontecer.

Identificador da conexão virtual Um VCI ou VPI.

Identificador de caminho virtual (VPI) Um campo do cabeçalho ATM que define a rota de roteamento.

Identificador de caminho virtual (VPI)/Identificador de canal virtual (VCI) Dois campos usados juntos para rotear uma célula ATM.

Identificador de canal virtual (VCI) Campo num cabeçalho de uma célula ATM que define o canal de transmissão.

Identificador de circuitos virtuais (VCI) Um campo no cabeçalho da célula ATM que define o canal.

Incumbent Local Exchange Carrier (ILEC) As companhias telefônicas que proviam os serviços antes de 1996 e possuíam um sistema de cabeamento próprio.

Institute of Electrical and Electronics Engineers (IEEE) Um grupo de profissionais especializados da engenharia elétrica e eletrônica, dividido em comitês, responsável pela divulgação de padrões (dentre outras coisas).

Integridade Indicador de qualidade da informação.

Intercalando dados Método para tomar uma quantidade específica de dados de cada dispositivo e uma ordem regular.

Interexchange Carrier (IXC) Estas companhias, às vezes denominadas companhias de longa distância, provêem serviços de comunicação entre dois consumidores situados em diferentes LATAs. Após o ato de 1996, estes serviços puderam ser provados por qualquer companhia, incluindo aquelas envolvidas apenas com os serviços intra-LATAs.

Interface A fronteira entre duas partes de um equipamento. Ela também refere-se às especificações mecânica, elétrica e funcional de uma conexão.

Hidden page

Hidden page

Hidden page

Hidden page

Network Allocation Vector (NAV) No método CSMA/CA, a quantidade de tempo necessário antes que uma estação verifique a ociosidade do meio.

Network Control Protocol (NCP) No protocolo PPP conjunto de protocolos que permitem o encapsulamento de dados oriundos dos protocolos da camada de rede.

Network Virtual Terminal (NVT) Protocolo da camada de aplicação do modelo TCP/IP que permite a operação de *logon* remoto.

Network-to-Network Interface (NNI) Nas redes ATM, a interface entre duas redes diferentes.

Nível de dados A quantidade de símbolos diferentes usados para representar um sinal digital.

Nó Um dispositivo de comunicação endereçável de uma rede (p.ex., computador ou roteador).

Nome do domínio No DNS, uma sequência de rótulos separados por pontos.

Nonce Número aleatório grande utilizado na distinção entre um pedido de autenticação novo ou uma tentativa de invasão através da repetição de um pedido.

NonReturn to Zero (NRZ) Codificação polar onde o nível de sinal é sempre positivo ou negativo.

NonReturn to Zero, Invert (NRZ-I) Codificação NRZ onde o nível de sinal é invertido toda vez que encontrado o *bit* 1.

NonReturn to Zero, Level (NRZ-L) Codificação NRZ onde o nível de sinal está diretamente relacionado ao valor do *bit*.

Normal Response Mode (NRM) Modo de comunicação no protocolo HDLC onde uma estação secundária deve receber permissão da estação primária antes que a transmissão prossiga.

Notação de barra Método abreviado para indicar a quantidade de 1s presentes na máscara.

Notação decimal com pontos Uma notação desenvolvida para facilitar a leitura dos endereços IP; cada byte é convertido para o seu equivalente decimal.

Notação hexadecimal com dois pontos Na versão IPv6, uma notação para o endereço IP consistindo de 32 dígitos hexadecimais, agrupados quatro a quatro e separados por dois pontos.

Núcleo A parte central de plástico ou vidro de uma fibra óptica.

Número de porta Um número inteiro que define um processo sendo executado em um *host*.

Número de porta efêmera Um número de porta usado pelo cliente.

Número de seqüência Número que indica a localização de um *frame* ou pacote numa mensagem.

O

Onda de rádio Ondas eletromagnéticas de energia na faixa de 3 kHz e 300 GHz.

Onda Infravermelha Uma onda cuja freqüência está situada entre 300 GHz e 400 THz; utilizada freqüentemente nas comunicações a curta distância.

Onda senoidal Representação de um sinal de tensão senoidal variável no tempo.

Open Shortest Path First (OSPF) Protocolo de roteamento interno baseado no estado do link.

Optical Carrier (OC) Hierarquia de portadoras definidas para a rede SONET. A hierarquia define 10 diferentes portadoras (OC-1, OC-3, OC-12,..., OC-192), cada qual suportando uma taxa de transmissão diferente.

Órbita A trajetória que um satélite percorre ao redor da Terra.

Orthogonal Frequency Division Multiplexing (OFDM) Método de multiplexação semelhante ao FDM, onde todas as subbandas são utilizadas por uma fonte em um dado tempo.

Overhead (sinalização) Bits extras agregados à informação com propósito de controle.

P

Pacote Sinônimo de unidade de dados; termo muito utilizado para o encapsulamento na camada de rede.

Pacote de atualização do estado do link Um pacote que fornece informação física sobre rotas ou roteares específicos.

Padrão da Internet Uma especificação da Internet completamente testada e aprovada. Ele é uma regulamentação formalizada tratando de algum aspecto da Internet.

Padrão de facto Um protocolo que ainda não foi aprovado por uma organização de padronização, mas que já foi adotado como padrão e se encontra bastante difundido.

Padrão de jure Um protocolo legislado oficialmente por uma organização de padronização internacional.

Página Unidade de hipertexto ou hipermídia disponível na Web.

Página da Web Unidade de hipertexto e hipermídia disponível na Web.

Paridade ímpar Método de detecção de erros que adiciona um bit extra à unidade de dados de forma que a soma de todos os bits 1s seja ímpar.

Paridade par Um método de detecção de erros no qual um bit extra é adicionado à unidade de dados de modo a tornar par a quantidade total de bits 1s.

- Partially Qualified Domain Name (PQDN)** Nome de domínio que não inclui todos os níveis da hierarquia de nomes entre o host e o domínio raiz.
- Passive open** Estado de um servidor onde ele espera a chegada de um pedido de um cliente.
- Password Authentication Protocol (PAP)** protocolo de autenticação simples utilizado pelo PPP.
- P-box** Um circuito de hardware usado para criptografar informação binária.
- Per Hop Behavior (PHB)** No modelo DiffServ, um campo de 6 bits que define o mecanismo de controle de pacotes.
- Período** Intervalo de tempo necessário para que uma onda complete um ciclo.
- Personal Communication System (PCS)** Termo genérico dado ao sistema comercial de telefonia celular que oferece muitos tipos de serviços de comunicação.
- P-frame** Um quadro MPEG que contém somente as modificações relativas ao quadro anterior.
- Phase Shift Keying (PSK)** Modulação por deslocamento de fase. Modulação na qual a fase da portadora varia de modo a representar um padrão de bits específico.
- Piconet** Rede Bluetooth.
- Piggybacking** Inclusão da mensagem de confirmação no frame de dados.
- Pilha dupla** Dois protocolos (IPv4 e IPv6) em uma estação.
- Pipelining** Na estratégia Go-Back-N ARQ, técnica onde muitos frames podem ser enviados antes do receptor dar alguma resposta sobre os frames anteriores.
- Pixel** Menor elemento de uma imagem.
- Placa de rede (NIC)** Dispositivo eletrônico, interno ou externo ao computador, responsável pela conectividade física, elétrica e funcional com a rede.
- Point Coordination Function (PCF)** Numa rede WLAN, método de acesso complexo implementado na infra-estrutura da rede.
- Polinômio** Termo algébrico que pode representar um divisor CRC.
- Política de roteamento** Característica encontrada no roteamento baseado no vetor de caminhos onde as tabelas de roteamento são baseadas em um conjunto de regras adotadas pelo administrador da rede.
- Poll** No método de acesso principal/secundário é um procedimento na qual a estação principal pergunta à estação secundária se ela deseja transmitir.
- Poll/select** Método de acesso ao protocolo usando os procedimentos poll e select. Veja Poll e Select.
- Ponto de presença (POP)** Ponto de intercomunicação entre as instalações de comunicações fornecidas pela empresa de telefonia e a instalação de distribuição principal do prédio.
- Pontos de sincronização** Pontos de referência introduzidos nos dados pela camada de sessão para propósitos de controle de fluxo e de erros.
- Porta** Num URL, número da porta de um servidor.
- Porta blocking** Uma porta de uma bridge que não encaminha frames.
- Porta conhecida** Número de porta que identifica um processo no servidor.
- Porta de encaminhamento** Uma porta de uma bridge que encaminha um frame recebido.
- Porta IrDA** Uma porta que permite a comunicação entre um teclado sem fios e um PC.
- Portadora** Um sinal de alta frequência usado na modulação analógica e digital. Uma das características da portadora (amplitude, frequência ou fase) é modificada de acordo com a informação modulante.
- Portadora comum** Uma facilidade de transmissão disponível ao uso público pelo órgão de regulamentação.
- Post Office Protocol, versão 3 (POP3)** Protocolo de acesso a e-mail mais simples e popular que o SMTP.
- Preâmbulo** Campo de 7 bits de um frame IEEE 802.3 consistindo de bits 1's e 0's alternados que informam ao receptor sobre a sincronização.
- Pretty Good Privacy (PGP)** Um protocolo que fornece todos os quatro aspectos de segurança para envio de e-mails.
- Privacidade** Aspecto de segurança na qual a mensagem faz sentido apenas para quem deve recebê-la.
- Problema n²** Problema provocado pela enorme quantidade de chaves simétricas que devem ser distribuídas.
- Processamento distribuído** Uma estratégia na qual os serviços fornecidos por uma rede residem em muitos locais diferentes.
- Processo** Um programa aplicativo rodando no cliente/servidor.
- Processo cliente** Um programa aplicativo rodando em um computador local que solicita os serviços de um programa aplicativo rodando em um computador remoto.
- Processo peer-to-peer** Processo das máquinas transmissora e receptora que se comunicam numa dada camada.
- Produto interno** Número produzido pela multiplicação escalar de duas sequências, elemento a elemento.
- Projeto 802** Projeto proposto pelo IEEE na tentativa de resolver a incompatibilidade entre LANs. Veja também Projeto IEEE 802.

Hidden page

Hidden page

Hidden page

- Segmento** (1) Pacote na camada TCP (2) Comprimento de um meio de transmissão compartilhado por vários dispositivos.
- Segurança** Proteção de uma rede de acesso não autorizado, vírus e catástrofes.
- Select** Procedimento segundo o qual uma estação primária pergunta a uma estação secundária se ela deseja receber dados.
- Selective-repeat ARQ** Método de controle de erros onde somente o frame de erro é retransmitido.
- Semântica** Significado de cada seção de bits.
- Separação horizontal** Método para melhorar a estabilidade do protocolo RIP no qual o roteador escolhe seletivamente a interface por onde a informação atualizada será enviada.
- Séries V** Padrões ITU-T que definem a transmissão de dados sobre as linhas telefônicas.
- Serviço analógico comutado** Uma conexão analógica temporária entre dois usuários.
- Serviço analógico dedicado** Um serviço que apresenta uma linha dedicada entre dois usuários.
- Serviço de área local** Um serviço telefônico que controla as chamadas locais.
- Serviço 900** Um serviço telefônico pago por quem efetua a discagem.
- Serviço 800** Um serviço telefônico sem custo para quem efetua a discagem.
- Serviço orientado à conexão** Um serviço para a transferência de dados envolvendo estabelecimento e término da conexão.
- Serviços de melhor esforço** Um mecanismo de transmissão não confiável usado pelo IP que não garante a entrega da mensagem.
- Serviços diferenciados (DS ou Diffserv)** Uma classe de serviços com QoS destinado ao protocolo IP.
- Serviços integrados (IntServ)** Uma classe de serviços com QoS destinado ao protocolo IP.
- Servidor** Programa provê serviços a outros programas (denominados clientes).
- Servidor concorrente** Um servidor que pode processar muitas solicitações ao mesmo tempo, compartilhando o tempo de execução entre as solicitações.
- Servidor concorrente orientado à conexão** Um servidor orientado à conexão que é capaz de servir muitos clientes ao mesmo tempo.
- Servidor de autenticação (AS)** O KDC no protocolo Kerberos.
- Servidor de mídia** Um servidor acessado pelo aplicativo media player durante as operações de download de arquivos de áudio/video.
- Servidor de registros** No protocolo SIP, um servidor que conhece os endereços IP das partes chamadas.
- Servidor DNS** Um computador que mantém informação sobre o espaço de nomes.
- Servidor iterativo sem conexão** Um servidor sem conexão que processa uma solicitação por vez.
- Servidor primário** Servidor que armazena toda a informação sobre uma zona de sua autoridade.
- Servidor proxy** Computador que mantém cópias das mensagens de resposta dos pedidos recentes.
- Servidor raiz** Servidor cuja zona consiste de toda a árvore. Um servidor raiz usualmente não armazena nenhuma informação sobre os subdomínios, mas delega autoridade a outros servidores, fazendo referência a eles.
- Servidor remoto** Programa executado em um site afastado fisicamente de um usuário.
- Servidor secundário** No esquema de DNS, servidor que faz backup de toda a informação sobre uma zona de outro servidor (primário ou secundário).
- Servidor TGS** Servidor Kerberos que utiliza os esquemas de tickets.
- Session Initiation Protocol (SIP)** Protocolo que estabelece, gerencia e termina uma sessão de multimídia nas aplicações de voz sobre IP.
- S-frame** Frame HDLC usado para funções de supervisão tais como confirmação, controle de fluxo e controle de erro; este frame não contém dados do usuário.
- Shielded Twisted-Pair (STP)** Cabo par trançado blindado.
- Short Interframe Space (SIFS)** No método CSMA/CA, o intervalo de tempo que o destino espera até receber o RTS.
- Simple Mail Transfer Protocol (SMTP)** Protocolo de aplicação TCP/IP que define o serviço de correio eletrônico na Internet.
- Simple Network Management Protocol (SNMP)** Protocolo TCP/IP que especifica o processo de gerenciamento da Internet.
- Sinal analógico** Uma forma de onda que varia continuamente com o tempo.
- Sinal aperiódico** Um sinal que não exibe padrão ou repetição de ciclo.
- Sinal composto** Um sinal composto de mais uma onda senoidal.
- Sinal digital** Um serviço oferecido por uma companhia telefônica caracterizado na hierarquia digital de sinais.
- Sinal periódico** Sinal que exibe um padrão repetitivo.
- Síndrome da janela boba** Situação na qual uma janela pequena é aberta pelo receptor e um segmento pequeno é enviado pelo transmissor.
- Sintaxe** A estrutura ou o formato dos dados, representando a ordem na qual eles devem ser processados.

Sistema aberto Modelo que permite a conexão entre dois sistemas de comunicação diferentes, independentemente da plataforma de hardware.

Slow start Método de controle de congestionamento segundo o qual o tamanho da janela de congestionamento diminui exponencialmente a princípio.

Socket Estrutura de software operando como um nó de extremidades de comunicações dentro de um dispositivo de rede.

Source Routing Bridge (SRB) Dentre as tarefas de uma SRB estão incluídas a filtragem, encaminhamento e bloqueio de frames. Num sistema utilizando SRB tais tarefas são realizadas pela estação de origem. A estação transmissora define as bridges que o frame deve visitar, evitando assim a existência de loops.

Spanning tree Uma árvore com o host de origem na raiz e grupos membros como folhas: uma árvore que conecta todos os nós da rede.

Spread spectrum Técnica de transmissão wireless que requer uma largura de banda muitas vezes maior que a largura de banda original do sinal.

Start Frame Delimiter (SFD) Campo de 1-byte no frame IEEE 802.3 que sinaliza o início da leitura da informação legível da cadeia de bits.

Stop-and-Wait ARQ Protocolo de controle de erros usando o mecanismo de controle de fluxo stop-and-wait.

Store-and-forward switch Switch que armazena o frame em um buffer de entrada até que todo o pacote tenha sido recebido.

Stream socket Estrutura projetada para ser utilizada com um protocolo orientado à conexão (como o TCP).

Streaming de áudio/vídeo armazenado Dados baixados como arquivos da Internet que um usuário pode escutar ou assistir.

Streaming de áudio/vídeo em tempo real Broadcast de dados da Internet por um usuário que pode escutar rádio ou assistir televisão.

Subcamada de convergência No protocolo ATM, a subcamada AAL superior que adiciona um cabeçalho ou trailer aos dados do usuário.

Subcamada de reconciliação Subcamada Fast Ethernet que passa dados no formato de quatro bits à camada MII.

Subcamada PHY O transceptor em uma rede Fast Ethernet.

Sub-rede Porção de uma rede.

Substituição monoalfabética Método de cifragem segundo o qual a cada ocorrência de um caractere é feita a substituição por outro caractere do alfabeto ou código utilizado.

Substituição polialfabética Método de criptografia no qual cada ocorrência de um caractere é substituído por um caractere diferente.

Substituição Método de cifragem baseado na substituição de caracteres onde um caractere é substituído por outro durante o processo.

Sufixo Para uma rede, a parte variável do endereço. No DNS, uma string usada por uma organização ou empresa para definir recursos ou hosts.

Supergrupo Sinal composto de cinco grupos multiplexados.

Supernet Rede formada por duas ou três rede menores.

Switch Dispositivo de conectividade que concentra muitas outras linhas de comunicação.

Switch de camada 2 Bridge com muitas portas, projetada para uma performance melhor.

Switch de camada 3 Switch que opera na camada de rede; um roteador.

Switched/56 Conexão digital temporária de 56 kbps entre dois usuários finais.

Switched-Ethernet Padrão Ethernet na qual um switch, substituindo um hub, pode dirigir a sua transmissão ao destino.

Symmetric Digital Subscriber Line (SDSL) Tecnologia DSL bastante similar à HDLC, mas usando somente um par de cabos trançados.

Synchronous Connection Oriented (SCO) Numa rede bluetooth, um link físico criado entre um mestre e um escravo que permite a reserva de slots em intervalos específicos.

Synchronous Digital Hierarchy (SDH) Padrão ITU-T equivalente a SONET.

Synchronous Optical Network (SONET) Padrão desenvolvido pela ANSI para transmissão de dados em alta velocidade através de fibras ópticas.

Synchronous Payload Envelope (SPE) Parte do frame SONET contendo os dados do usuário e a sinalização (overhead) da transmissão.

Synchronous Transport Signal (STS) Um sinal na hierarquia SONET.

T

Tabela de roteamento Tabela contendo informação que um roteador necessita para rotear pacotes. A informação pode incluir o endereço de rede, o custo, os endereços do próximo salto e assim por diante.

Tag Instrução de formatação de documentos inseridos no HTML.

Tamanho do excesso de rajada (B_e) Em uma rede Frame Relay, a quantidade máxima de *bits* que excede a taxa B_s , que um usuário pode transmitir durante um intervalo de tempo predefinido.

Tamanho máximo da rajada (B_{\max}) A quantidade máxima de *bits*, em um período de tempo específico, que uma rede Frame Relay deve transferir sem realizar descartes de pacotes.

Tamanho máximo da rajada Refere-se ao intervalo de tempo que o tráfego pode acontecer à taxa máxima.

Taxa constante de *bits* (CBR) A taxa de dados de uma classe de serviços ATM que foi projetada para aplicações que requerem áudio/video em tempo real.

Taxa de acesso Numa rede Frame Relay, a taxa de dados que nunca pode ser excedida.

Taxa de amostragem Número de amostras obtidas por segundo em um processo de amostragem.

Taxa de *bits* (bit rate) Número de *bits* por segundo.

Taxa de dados de pico O maior valor que um tráfego de dados pode atingir.

Taxa de modulação Quantidade de modulações (sinalizações) realizadas durante 1 s. A taxa de modulação é medida em *band*.

Taxa de pulsos Número de símbolos por segundo.

Taxa de transmissão Número de *bits* por segundo de uma rede.

Taxa não especificada de *bits* (UBR) Taxa de dados de um serviço ATM que garante apenas o serviço de melhor esforço.

Taxa variável de *bits* (VBR) Taxa de dados de uma classe de serviços ATM para usuários que necessitam de uma taxa variável de *bits*.

TDM bus Comutação por divisão do tempo na qual as linhas de entrada e saída estão conectadas a um barramento de alta velocidade através de microchaves.

Telecomunicações Troca de informação à distância utilizando equipamentos eletrônicos.

Teleconferência Comunicação de áudio e vídeo entre usuários remotos.

Teledesic Rede de satélites para prover comunicação semelhante às fibras ópticas (canais banda larga, com baixas taxas de erros e pequenos atrasos).

TELNET Veja *Terminal Network*.

Tempo de *bit* O tempo necessário para transmitir um *bit*.

Tempo de propagação Tempo necessário para que um sinal viaje de um ponto a outro em uma rede.

Teorema de Nyquist Teorema que afirma que a quantidade de amostras para representar adequadamente um sinal analógico é igual a duas vezes o valor da componente de frequência mais alta do sinal original.

Terminador Dispositivo eletrônico que evita a reflexão do sinal no final do cabo.

Terminal Network (TELNET) Programa cliente-servidor de propósito geral que permite *login* remoto.

Término da conexão Uma mensagem enviada com objetivo de finalizar a conexão.

Teste de paridade Método de detecção de erros que utiliza o *bit* de paridade.

Tetrabit Unidade de dados formada por quatro *bits*.

Texto cifrado Os dados criptografados.

Texto limpo Mensagem original sem criptografia.

thick Ethernet Veja *10Base5*.

thin Ethernet Veja *10Base2*.

Throughput Quantidade de *bits* que passam em um ponto por segundo.

Ticket Mensagem cifrada contendo uma chave de sessão.

Time Division Duplexing TDMA (TDD-TDMA) Em uma rede *Bluetooth*, um tipo de comunicação *half-duplex* onde o escravo recebe e envia dados, mas não ao mesmo tempo.

Time-Slot Interchange (TSI) Um método TDM consistindo de uma memória RAM e uma unidade de controle.

Timestamp Uma opção do cabeçalho IP utilizada para registro de tempo de processamento de um *frame* pelo roteador. Além disso, *timestamp* também é um método de controle de *jitter* no tráfego interativo de áudio e vídeo em tempo real.

Time-To-Live (TTL) Tempo de vida de um pacote.

Tipos de serviços (ToS) Critério ou valor que especifica o controle do datagrama.

Token Pacote pequeno utilizado no método de acesso de passagem da permissão.

Token bucket Algoritmo que possibilita aos *hosts* ociosos acumularem crédito (na forma de fichas) para uso futuro.

Token passing Método de acesso onde um *token* (permissão) circula pela rede. A estação que captura o *token* pode transmitir dados.

Token Ring LAN utilizando topologia em anel e método de acesso de passagem permissão.

Topologia A estrutura de uma rede incluindo a organização física dos equipamentos.

Topologia barramento Uma topologia na qual todos os computadores compartilham um meio comum (um único cabo).

Hidden page

Hidden page

Índice

- 1000Base-SX, [322-323](#)
1000Base-CX, [322-323](#)
1000Base-LX, [322-323](#)
1000Base-T, [322-324](#)
1000Base-X, [322-324](#)
100Base-FX, [318-320](#)
100Base-T4, [318-322](#)
100Base-TX, [318-320](#)
100Base-X, [318](#)
10Base2, [312-313](#)
10Base5, [312](#)
10Base-FI, [313-314](#)
10Base-T, [312-313](#)
16-QAM, [137-139](#)
1-persistente, [292-293](#)
2B1Q, [110-111](#)
2-PSK, [135-136](#)
3-slots frame, [342-343](#)
4B/5B, [113-114](#) 320-321
4D-PAM5, [323-324](#)
4-PSK, [136](#)
4-QAM, [137-139](#)
802.11, [336-337](#)
8B/10B, [113-114](#), [323-324](#)
8B/6T, [114-115](#)
8-PSK, [136](#)
8-QAM, [137-139](#)
- A**
- AAL, [408-409](#)
AAL1, [409-410](#)
 CS, [409-410](#)
AAL2, [408-411](#)
 camadas CS, [410-411](#)
AAL3/4, [410-411](#)
AAL5, [412-413](#)
ABM, [265-266](#)
Abordagem de circuito virtual, [426-427](#)
Abordagem de datagrama, [426-428](#)
Acesso aleatório, [280](#)
Acesso ao meio, [289](#)
 Gigabit Ethernet, [321-322](#)
 métodos, [289](#)
Acesso controlado, [294-295](#)
Acesso múltiplo; *veja* MA
Acesso ponto a ponto, [277-278](#)
ACK, [254-255](#)
 mechanism *Stop-and-Wait* ARQ, [255](#)
 poll, [295-296](#)
ACL, [342-343](#)
Address Resolution Protocol; *veja* ARP
- ADSL, [213-214](#), [216-217](#)
 adaptativa, [214](#)
 conexão local, [213-214](#)
 DMT, [214-215](#)
 HDSL, [215-216](#)
 taxa de transmissão real, [214-215](#)
 VDSL, [216-217](#)
Advanced Mobile Phone System; *veja* AMPS
Advanced Research Projects Agency; *veja* ARPA
AF PHB, [577-578](#)
Agências reguladoras, [48-49](#)
Agendamento, [560](#)
 fila de prioridade, [560-570](#)
 fila FIFO, [560-570](#)
 WFQ, [560-570](#)
Algoritmo de Bellman-Ford, [485-486](#)
Algoritmo de Dijkstra, [494-495](#), [509](#)
 árvore de custo mínimo, [509](#)
 BGP, [496](#)
Algoritmo de Karn, [551-552](#)
 AS, [725-726](#)
 KDC, [731](#)
Algoritmo de Nagle, [547](#)
Algoritmo *spanning tree*, [354-355](#)
Alocação de buffer, [411-412](#)
ALOHA, [289-290](#)
Alternate Mark Inversion; *veja*AMI
AMI, [145-149](#)
American National Standards Institute; *veja* ANSI
American Standard Code for Information Interchange; *veja* ASCII
AMI, [109-111](#), [215-216](#)
Amostra e mantém (*sample and hold*), [115-116](#)
Amostragem, [114-116](#)
 PAM, [115-116](#)
Amplificador, [350](#)
 atenuação, [92-93](#)
 TV a cabo, [216-217](#)
Amplitude, [129-132](#)
 ASK, [131](#)
 FM, [147-149](#)
 FSK, [133](#)
 medição, [75](#)
 onda senoidal, [77-78](#)
 PM, [149-150](#)
 PSK, [135-136](#)
 QAM, [137-139](#)
Amplitude de pico, [75](#)
- Amplitude Modulation; *veja* AM
Amplitude Shift Keying; *veja* ASK
AMPS, [371-372](#)
Análise de Fourier, [80-81](#)
Anel, [39-41](#)
 definição, [40-41](#)
 desvantagens, [42](#)
 duplo, [42](#)
 repetidor, [40-41](#)
 vantagens, [42](#)
Ângulo crítico, [180-181](#)
Ângulo de incidência, [180-181](#)
ANSI, [35](#), [47-49](#)
Antena
 corneta, [187-189](#)
 direcional, [185-186](#)
 foco, [187-189](#)
 omnidirecional, [187-188](#)
 prato parabólico, [187-189](#)
 satélite, [380-381](#)
 unidirecional, [187-189](#)
Antena comunitária de TV, [216-217](#)
AP, [329](#), [336-337](#)
Apller, [656-657](#)
Application Adaptation Layer; *veja* AAL
Área, [488-489](#)
Aritmética em complemento de um, [242](#)
ARP, [458-459](#)
 campo operação, [459-460](#)
 componentes do pacote, [458-459](#)
 comunicação *host-to-host*, [461-462](#)
 encapsulamento, [460-461](#)
 formato do pacote, [450-460](#)
 mapeamento endereço IP em endereço físico, [458-459](#)
 mensagem de consulta (*query*) de broadcast, [458-459](#)
 operação, [460-461](#)
 pacote de consulta (*query*), [458-459](#)
 quatro casos, [460-461](#)
 tamanho do endereço físico, [459-460](#)
 tamanho do protocolo, [459-460](#)
 tipo de hardware, [459-460](#)
 tipo de protocolo, [459-460](#)
ARPA, [44-45](#)
ARPANET, [44-45](#)
ARQ, [254-255](#)
 produz banda × atraso de propagação, [264-265](#)
Arquitetura cliente-servidor, [591](#)
 conceito de processo, [593-594](#)
 concorrente, [592-593](#)

- endereçamento, 588-589
 programas aplicativos, 592-593
 Arquitetura da Internet, 55-56, 760
 cabeçalho, 57
 camada de aplicação, 64-65, 587-588, 762-763
 camada de enlace, 229-230
 camada de rede, 59-60
 camada de transporte, 61-62
 camada física, 56, 58
 camadas de suporte à rede, 57
 camadas de suporte ao usuário, 57
 camadas, 760
 comunicação em camadas, 55-56
 interface entre camadas, 56
 organização, 57
 processos peer-to-peer, 56
trailer, 57
 Arquivo de host, 603-604
 Arquivo de zona, 606-607
 Árvore baseada na fonte, 503-504
 Árvore compartilhada no grupo CBT, 510-511
 Árvore de caminho mais curto, 494-495
 Árvore de custo mínimo, 500
 Árvore multicast compartilhada, 503-505
 AS, 484, 734-735
 ASCII, 35, 749
 estendido, 35
 ASK, 129-131, 133
 com PSK, 137-139
 conceito, 131-132
 exemplo de largura de banda, 132-133
 exemplo de taxa de transmissão, taxa de modulação, 132-133
 fórmula para a largura de banda, 131-132
 largura de banda, 131-132
 portadora, 131
 QAM, 138-139
 ruído, 131-132
 ASN, 31-32, 734
 Assinatura digital, 712-713
 assinando a síntese, 713-714
 assinando todo o documento, 712-713
 autenticação, 713-714
 função de hash, 713-714
 integridade, 713-714
 não repúdio, 713-714
 PGP, 739-740
 Asymmetrical DSL; *veja* ADSL
 Asynchronous Balanced Mode; *veja* ABM
 Asynchronous Connectionless Link; *veja* ACL
 Asynchronous Transmission mode; *veja* ATM
 AT&T Bell System, 206-207
 Ataque *bucket brigade*, 720-721
 Ataque *man-in-the-middle*, 720-721
 Ataque *playback*, 769-770
 Ataque por reflexão, 717-718
 Ataque *replay*, 716-717
 KDC, 720-721
 Atenuação, 91-92, 540
 amplificador, 92-93
 exemplo, 92-93
 fibra óptica, 184-185
 ATM, 401-403-406
 AAL1, 409-410
 AAL2, 410-411
 AAL3/4, 410-411
 AAL5, 412-413
 arquitetura, 403-404
 atraso de transferência de células, 582
 atributos relacionados à rede, 582
 atributos relacionados ao usuário, 581-582
 cabeçalho NNI, 407-408
 camada, 406-407
 camada ATM, 407-408
 camada física, 407-408
 célula, 405-406
 classes de taxas de transmissão disponíveis, 581-582
 compatibilidade reversa, 401
 comutação, 405-406
 conexão virtual, 403-404
 conexão, 405-406
 estabelecimento da conexão, 405-406
 estrutura da comutação, 405-406
 exemplo, 405-406
highways da informação, 401
 identificador, 404-405
 liberação da conexão, 405-406
 metas do projeto, 401
 multimídia, 775
 multiplexação, 402-403
 orientado à conexão, 401
 QoS, 580-581
 roteamento hierárquico, 405-406
 SONET, 406-407
 SVC, 405-406
 taxa constante de bits, 580-581
 taxa de erro de células, 582
 taxa de perda de células, 582
 taxa máxima de células, 581-582
 taxa mínima de células, 582
 taxa não especificada de bits, 581-582
 taxa sustentável de células, 581-582
 taxa variável de bits, 580-582
 TDM assíncrono, 402-403
 tipos de conexão, 405-406
 tolerância à variação do atraso de células, 582
 variação do atraso de células, 582
 WAN, 775
 ATM LAN, 412-413, 775
 arquitetura mista, 776-777
 arquitetura, 776
 BUS, 778-779
 cliente/servidor, 777-778
 expansão, 775
 LANE, 776-777
 legada, 776
 pura, 776
 vantagens, 776-777
 Atraso, 561-562
 carga, 562-563
 comutação por divisão de tempo, 200-201
 tempo real, 685-686
 Atraso da resposta de confirmação (ACK), 548
 Atraso de propagação
 CSMA, 291-292
 LEO, 384
 Atraso de resposta, 502-503
 Atributos de caminho, 496-497
 AS_PATH, 497-498
 não transitivo, 497-498
 NEXT-HOP, 497-498
 origem, 496-497
 transitivo, 497-498
 Attachment Unit Interface; *veja* AUI
 Áudio em tempo real, 675-676
 Áudio/vídeo em tempo real, 675-676
 Áudio/vídeo interativo, 675-676
 Áudio/vídeo interativo em tempo real, 674-675
 Áudio/vídeo sob demanda, 665
 AUI, 310-311
 MII, 318
 Autenticação, 278-279, 281-282, 470-471, 721-723, 726-727
 assinatura digital, 723-724
 bidirecional, 728
 pacotes, 728-729
 protocolo AH, 735
 Autenticação de entidades, 715-716
 Automatic Repeat Request; *veja* ARQ
 Auto-negociação, 312
 Autonomous System, 484, 487-488
 área, 488-489
 backbone, 488-489
 roteamento baseado no vetor de caminhos, 496
 Auto-sincronização, 106
- B**
- B, 578-579
 rajada de dados, 580-581
 B, 579-580
 rajada de dados, 580-581
Back pressure, 564-565
Backbone, 488-489
 comutado, 350-360
 estrela, 350-360
 identificador de área, 488-489
 linear, 40-41
link virtual, 488-489
Backoff, 293-294
 ALOHA, 290-291
 Backward Explicit Congestion Notification; *veja* BECN
 Banco de dados estático, 441-442
 Banda
 AMPS, 371-372
bluetooth, 340-341
 D-AMPS, 372-373
 DS-SS, 331-332
 FHSS, 331-332
 GSM, 373-374
 HR-DS-SS, 332-333
 IS-95, 374-376
 OFDM, 332-333
 Banda de segurança, 155-156
 grupo jumbo, 159
 sistema de telefonia, 159
 Bandas, 186-187
 Base 16, 755-756

- Base 8, 754-755
 Base de dados (*database*), DHCP, 441-442
 Base de dados de *link state*, 493-494
 algoritmo Dijkstra, 494-495
 Base de dados dinâmica, 441-442
 Basic Service Set; *veja* BSS
 BASize, 411-412
 Batcher, 786 (773-774)
 Baud, 139-140
 BECN, 399-400
 mecanismo, 567-568
 transmissor, 567-568
 Bell Operating Company; *veja* BOC
 B-frame, 681-682
 BGP 484-485, 495-496
 mensagem *keepalive*, 497-498
 mensagem *notification*, 497-498
 mensagem *open*, 497-498
 mensagem *update*, 497-498
 porta, 787
 roteamento baseado no vetor de caminhos, 495-496
 tipos de mensagens, 497-498
 Bipolar n-Zero Substitution; *veja* BnZS
 Bit, 117-119
 baud, 139-140
 comparação taxa de modulação (*baud rate*), 139-140
 relação de fase, 136
 Bit de início (*start bit*), transmissão assíncrona, 120-121
 Bit de parada (*stop bit*), transmissão assíncrona, 120-121
 Bit de paridade, 235-236
 código Hamming, 246
 Bit de redundância, 245-246
 Bit mais fragmentos, 466-467
 Bit não fragmentar, 466-467
 Bit P/F, 268
 Bit URG, 553
 Bits de enchimento (*bit padding*), 164-165
 Bits de enchimento (*bit stuffing*), 272-273
 exceções, 272-273
 HDLC, 271-273
 TDM, 164-165
 Bits de enchimento (*framing bit*), 164
 Bits por segundo, 84-85
 Block descriptor, 633-634
 Blocking, 198-199
 Bloco produto, 698-699
 Bluetooth, 329, 338-339
 aplicações, 338-339
 arquitetura, 339-340
 camadas, 340
 dispositivos, 340
 formato do frame, 342-343
 Bluetooth LAN, 339-340
 BnZS, 109-110
 BOC, 206-207
 Border Gateway Protocol; *veja* BGP
 Bridge, 313-314, 352
 como um filtro, 352
 conectando LANs, 352-353, 356-357
 designada, 355-356
 dinâmica, 352-353
 domínio de colisão, 314-315
 Ethernet, 313-314
 função, 313-314, 352
 múltiplas LANs, 356-357
 problema do loop, 354-355
 redundante, 354-355
 remota, 359-360
 source routing, 356-357
 switch de camada 32, 357-358
 transparente, 357-358
 Bridged Ethernet, 313-314
 Bridge Protocol Data Unit (BPDU), 356-357
 Broadcast/Unknown Server; *veja* BUS
 Broadcasting
 VLAN, 361-362
 Browser, 640, 649-651
 arquitetura, 640
 componentes, 640
 controlador, 640
 documento dinâmico, 653-654
 HTML, 649-651
 interpretador, 640
 Markup Language, 650-651
 programa cliente, 640
 BSS, 329
 Brag, 411-412
 Buffer
 circular, 535
 controle de fluxo, 253-254
 receptor, 535-536
 roteador, 784
 servidor, 596-597
 TCP, 535, 544-545
 transmissor, 535-536
 Buffer de reprodução, 677-678
 Bus, 39-41
 BUS, 791-794
 Byte, 428-429
 Byte de sincronização, 122-123
 Byte de urgência, 553
- C**
- CA, 704-705, 723-725
 Cabeça, 216-217
 Cabeça de rede, 216-217
 Cabeça de rede regional, 217-218
 Cabeçalho, 57
 camada de transporte, 522
 célula, 405-406
 Cabeçalho base, 473
 Cable Modem Transmission System; *veja* CMTS
 Cable modem; *veja* CM
 Cabo, par trançado, 176
 Cabo coaxial, 176, 178-181
 aplicações, 180-181
 blindagem, 178-179
 condutor, 178-179
 conector, 179-180
 estrutura, 178-179
 Ethernet, 180-181
 faixa de frequência, 178-179
 HFC, 217-218
 padrões, 179-180
 performance, 179-180
 rede telefônica, 179-180
 taxas RG, 179-180
 TV a cabo, 180-181, 216-217
- Cabo de fibra óptica, 176
 truncamento, 202-203
 Caching, 612-613
 Cadeia de bits, 69-70
 Caixa de correio, 620
 Camada ATM, 406-407
 cabecalho UNI, 407-408
 campo VCI, 408-409
 campo VPI, 408-409
 CLP, 408-409
 controle de congestionamento, *veja* CLP
 controle de fluxo genérico, *veja* GFC
 correção de erro do cabecalho, 408-409
 formato do cabecalho, 407-408
 função, 407-408
 GFC, 408-409
 nível de controle de fluxo NNI, 408-409
 nível de controle de fluxo UNI, 408-409
 tamanho da célula, 407-408
 tipo de payload, 408-409
 VPI, 408-409
 Camada banda base, 340-341
 Camada de aplicação, 64-65, 587-588, 762-763
 acesso à Web, 64-65
 camada de transporte, 521
 correo eletrônico, 64-65, 762-763
 funções, 762-763
 login remoto, 64-65
 manipulação de arquivos, 64-65, 762-763
 modelo da Internet, 763-764
 NVT, 762-763
 PGP, 739-740
 segurança, 733
 serviços de diretório, 763-764
 serviços, 762-763
 Camada de apresentação, 65-66, 761
 cifragem, 761-762
 compressão, 762-763
 funções, 761
 tradução, 761
 Camada de enlace, 58, 229-230
 ação local, 229-230
 controle de acesso, 59
 controle de erro, 59, 230, 253-255
 controle de fluxo, 59, 230, 253-254
 endereçamento, 59
 endereçamento físico, 59
 framing, 59
 função, 58
 modelo da Internet, 229-230
 serviços, 229-230
 subcamadas, 230-231
 Camada de rádio, 340-341
 Camada de rede, 59-60, 419, 423-424
 endereçamento lógico, 59-60
 endereço, 420
 endereço múltiplo, 420
 na origem, 423-424
 no destino, 425-426
 pacote, 59-60
 roteamento, 60-61, 420-421
 serviços, 420
 Camada de sessão, 65-66, 761
 Camada de transporte, 61-62
 cabecalho, 522

- camada de aplicação, 521
com conexão, 521
controle da conexão, 62-63, 522
controle de erros, 62-63, 522-523
controle de fluxo, 62-63, 522-523
demultiplexação, 528-529
endereçamento, 522-523
estabelecimento da conexão, 522-523
funções, 61-62, 521
geração de pacotes, 522
mensagem longa, 522
multiplexação, 528-529
ordenando os datagramas, 427-428
orientada à conexão, 62-63
porta de endereçamento, 62-63
remontagem, 62-63
segmentação, 62-63
segurança, 737-738
sem conexão, 62-63
serviços, 522
tráfego em tempo real, 687-688
transmissão orientada à conexão, 522-523
- Camada física**, 58, 69-70
ATM, 406-407
Ethernet, 310-311
Fast Ethernet, 317
Frame Relay, 398-399
função, 58
Gigabit Ethernet, 321-322
meios de transmissão, 175
modelo da Internet, 56
propósito, 58
representação dos bits, 58
sinais, 73
sincronização dos bits, 58
tarefas, 69-70
taxa de dados, 58
- Camadas de suporte à rede**, 57
- Camadas de suporte ao usuário**, 57
- Camadas superiores do modelo OSI**, 95-96
- Campo AL**, 411-412
- Campo controle**
HDLC, 267
tipos, 268
- Campo flag**, 266-267
- Campo informação HDLC**, 267
- Campo protocolo**
CHAP, 282-283
pacote PAP, 281-282
protocolo AH, 734-735
- Campo tamanho**, 411-412
- Canal**, 38-39, 155-156
- Canal com ruído**, 90-91
- Canal livre de ruído**, 89-90
- Canal piloto**, 376-377
- Canalização**, 296-297
- Capacidade da rede**, 562-563
- Capacidade de Shannon**, 90-91, 91-92
- Capacidades**, 167-168
- Características do fluxo de dados**, 568-569
- Carga**, 561-562
- Carrier Sense Multiple Access with Collision Avoidance**; *veja* CSMA/CA
- Carrier Sense Multiple Access with Collision Detection**; *veja* CSMA/CD
- Carrier Sense Multiple Access**; *veja* CSMA
- Casca**, 181-182
- CATV**, 216-217
- CBT**, 509-510
Autonomous System, 509-510
deixando o grupo, 509-510
encapsulamento, 510-511
pacote *multicast*, 510-511
roteador *core*, 510-511
roteador *rendezvous*, 509-510
- CCITT**, 47-48
- CCK**, 332-333
- CDMA**, 296-298, 374-379
banda larga, 379-380
codificação, 298-299
demultiplexador, 299-300
DSSS, 331-332
geração da sequência, 299-300
multiplexador, 298-299
- CDMA2000**, 379-380
- Célula**, 230, 402-403, 405-406, 582
ATM, 405-406
cabecalho, 405-406
definição, 402-403
estrutura, 405-406
payload, 405-406
tamanho, 405-406
- Central de comutação**, 201-203
POP, 203-204
- Central local**, 202-203
- Cerf. Vint**, 44-45
- CGI**, 653-654
- Challenge**, 281-282
- Challenge Handshake Authentication Protocol**; *veja* CHAP
- Chamada local**, 204-205
- CHAP**, 281-283
- Chave**, 694, 694-695
- Chave de sessão**, 716-717, 720-721
Diffie-Hellman, 718-719
TGS, 726-727
única, 721-723
- Chave privada**, 703-704, 711-712, 705-706
PGP, 739-740
RSA, 704-705
- Chave pública**, 703-704, 711-712, 718-719, 723-724
RSA, 704-705
- Chave secreta**, 694
PGP, 739-740
- Chave simétrica**, 694, 715-716, 718-721
Diffie-Hellman, 720-721
- Cheapernet**; *veja* 10Base2
- Chip**, 297-298
- Chip code**, 331-332
- Choke packer** (pacote de alerta), 564-565
- Ciclo**, 73-74
fase, 77-78
infinito, 76-77
- CIDR**, 441-442, 449-450
algoritmos de procura da tabela de roteamento, 450-451
entradas da tabela, 449-450
- IPv6**, 472-473
roteamento geográfico, 450-451
roteamento hierárquico, 450-451
- tamanho da tabela de roteamento, 449-450
- Cifra de César**, 695-696
- Cifra de Vigenère**, 696-697
- Cifragem**, 470-471, 693-694
camada de apresentação, 762-763
- DES**, 698-699
- monoalfabética**, 695-696
- RSA**, 704-706
- Cifras**, 694
por bloco, 698
por substituição, 694-695, 781-782
por transposição, 707
- Cinturão de Van Allen**, 381, 382-383
- Cipher Block Chaining (CBC)**, 702
- Cipher Feedback Mode (CFM)**, 702-703
- Cipher Stream Mode (CSM)**, 702-703
- CIR**, 579-580
- Círculo virtual**, IntServ, 573-574
- Círculo virtual comutado**; *veja* SVC
- Círculo virtual permanente**; *veja* PVC
- Circuitos de comutação**, 772-773
- Circuitos virtuais**; *veja* VP
- Classe A**, 431-432
- Classe B**, 432-433
- Classe C**, 433-434
- Classe D**, 434
- Classe de serviços**, 574-575, 580-581
carga controlada, 574-575
garantido, 574-575
- Classe E**, 434
- Classes de fluxo**, 569
- Classless InterDomain Routing**; *veja* CIDR
- Clear To Send (CTS)**, 333-335
- CLEC**, 203, 207-208
- Cliente**, 526-527, 591-592, 592-593, 596-597
chamada de escrita, 597-599
chamada de leitura, 597-599
concorrente, 592-593
concorrente orientado à conexão, 597-599
conexão, 597-599
criação do socket, 596-597
definição, 592-593
iterativo, 592-593
mensagem *active close*, 592-593
mensagem *active open*, 592-593
repetição, 596-597, 597-599
- Cliente LANE**; *veja* LES
- CM**, 213-214, 219
- CMTS**, 219
- Coax**; *veja* Cabo coaxial
- Code Division Multiple Access**; *veja* CDMA
- Codificação**
1000Base-X, 323-324
100Base-FX, 319-320
100Base-TX, 319-320
AMI, 109-110
bifásica, 109-110
bipolar, 106, 109-110
Manchester, 107-110, 310-311
Manchester diferencial, 109-110
NRZ, 107-108
NRZ-I, 107-108
NRZ-L, 107-108
polar, 106-108, 129-130

- unipolar, 106-107, 113-114
 Codificação de blocos, 111-112
 4B/5B, 113-114
 8B/10B, 113-114
 8B/6T, 114-115
 codificação de linha, 112-113
 controle de erro, 112-113
 divisão, 112-113
 etapas, 112-113
 substituição, 112-113
 Codificação de linha, 103-104
 aspectos, 103-104
 codificação de blocos, 112-113
 Codificação perceptiva, 677-678
 Codificação preditiva, 677-678
 Codificação *trellis*, 142-144
 Código, 35
 Código ASCII, 233-234
 Código de correção de erros, 244-245
 Código Hamming, 245-246
 Código 8B/6T, 765
 full-duplex, 315-316
 Códigos
 domínio geográfico, 609-610
 domínio organizacional, 608-609
 Colisão, 280
 CSMA, 291-292
 MAC, 230-231
 tempo de *backoff*, 297-298
 wireless, 335-336
 Common Gateway Interface; *veja CGI*
 Competitive Local Exchange Carriers; *veja CLEC*
 Complementary Code Keying; *veja CCK*
 Complemento, matriz de Walsh, 300-301
 Componente DC, 106-107
 Componentes do pacote de resposta, 458-459
 Compressão
 espacial, 670-671
 FTP, 633-634
 MPEG, 670-671
 temporal, 670-671
 Compressão de dados, camada de apresentação, 761
 Comprimento, 95-96
 Comunicação
 fonte-destino, 59-60
 host-to-host, 419
 process-to-process, 61-62
 Comunicação, aspectos, 253-254
 Comunicação com um escravo, 340-341
 Comunicação de dados, 33-34
 Comunicação entre escravos, 341-342
 Comunicação *multicast*, 431-432
 Comunicação processo a processo UDP, 532-533
 Comunicação roteador-*host*, 461-462
 Comunicação sem fios, 184-185
 Comunicação *unicast*, 431-432
 Comunicação via satélite, 379-380
 Comutação de células, 401
 Comutação de circuitos, 195-196, 391
 entradas e saídas, 196
 exemplo, 195-196
 redução do *link*, 195-196
 Comutação de circuitos virtuais, 391
 confirmação, 395-396
 fases, 392
 Comutação de mensagens, 195-196
 Comutação de pacotes, 195-196, 391, 425-426
 abordagem de circuito virtual, 426-427
 abordagem de datagrama, 427-428
 abordagens, 426-427
 IP, 462-464
 Comutação híbrida, 201-202
 Comutação por divisão de espaço, 200-201
 Comutação por divisão de tempo, 200-201
 Comutador Batcher-Banyan, 773-774
 Comutador de um estágio
 blocking (bloqueio), 198-199
 necessidade dos pontos de cruzamento, 198-199
 Comutador matricial, 196-197, 772-773
 Comutador multiestágio, 196-197, 201-202
 banyan, 772-773
 blocking, 198-199
 caminhos múltiplos, 197-198
 comparado ao comutador matricial, 196-197
 comutador intermediário, 197-198
 considerações de projeto, 196-197
 necessidade de pontos de cruzamento, 198-199
 primeiro estágio, 197-198
 Space-Time-Time-Space (STTS), 201-202
 terceiro estágio, 197-198
 Time-Space-Space-Time (TSST), 201-202
 Time-Space-Time (TST), 201-202
 Comutador por divisão de espaço, 196-197
 Comutador por divisão de tempo, 196-199
 Conceito de vizinhança, roteamento baseado no vetor de caminhos, 496
 Concorrência, 502-503
 Concorrente orientado à conexão
 comunicação, 506-507
 servidor, 503-504, 506-507
 Condicionamento do tráfego, 570-571
 Condutor
 par trançado, 176
 transmissão não guiada, 184-185
 Conector, 177-179
 cabô coaxial, 179-180
 fibra óptica, 183
 Conector BNC, 179-180
 Conector SC, 183
 Conector ST, 183
 Conexão, 38-39, 539-540
 Conexão local, 201-203
 ADSL, 213-214
 central de comutação, 202-203
 conexões, 203-204
 filtro, 214
 largura de banda, 214
 sinal, 204-205
 Conexão não persistente, 297-298
 Conexão persistente, 647-648
 Conexão ponto a ponto, 195-196
 Confiabilidade, 37-38, 568-569
 Confiabilidade da camada de rede, 531-532
 Confidencialidade, 211
 Confirmação 395-396, 548-549
 ALOHA, 290-291
 controle de fluxo, 230, 253-254
 Go-Back-N, 260
 Confirmação negativa; *veja NAK*
 Confirmação perdida, 549-550
 Congestionamento, 399-400, 561-562, 567-568
 conceito, 564-565
 evitando, 565-566
 exemplo, 561-562
 fila, 561-562
 Frame Relay, 399-400
 leaky bucket, 570-571
 método *additive increase*, 566-567
 método *multiplicative decrease*, 566-567
 prevenção, 563-564
 retransmissão, 565-566
 Constelação, 135-136
 Consulta DNS, 613-614
 Consulta ao ponteiro, 609-610
 Consulta ao roteador, 502-504
 Consulta inversa, 609-610
 Consultative Committee for International Telegraphy and Telephony; *veja CCITT*
 Contagem de saltos, 484
 Controlador, 40, 649
 Controle da conexão, 62-63, 522
 Controle de acesso, 59
 Controle de acesso ao meio; *veja MAC*
 Controle de admissão, 573-574
 Controle de congestionamento, 401, 552-553, 559, 561-565
 Frame Relay, 567-568
 malha aberta (sem realimentação), 563-564
 malha fechada (com realimentação), 564-565
 papel da rede, 565-566
 papel do receptor, 565-566
 Controle de diálogo, 65-66, 761
 Controle de erro, 59, 62-63, 253-255, 522-523, 548
 ACK perdido, 549-550
 camada de enlace, 230, 254-255
 camada de transporte, 62-63
 codificação de blocos, 111-112
 conceito, 253-254
 HDLC, 268
 retransmissão, 254-255
 segmento duplicado, 548-549
 segmento fora de ordem, 549-550
 segmento perdido, 548-549
 timers, 548-549
 Controle de fluxo, 59, 62-63, 253-255, 522-523, 543-544
 buffer, 253-254
 camada de enlace, 230, 253-254
 camada de transporte, 62-63
 conceito, 253-254
 Frame Relay, 566-567
 HDLC, 268
 janela deslizante, 543-544
 no protocolo IP, 468-469
 receptor, 253-254
 Controle de tráfego
 Frame Relay, 578-579
 PVC, 578-579
 SVC, 578-579
 Controle do link lógico; *veja LLC*
 Conversão analógico-analógico, 145

- Conversão analógico-digital, 114-115
Core, 181-182, 509-510
Core-Based Tree; *veja CBT*
Corpo, 650-651
Correção de erros, 233-234, 244-245
Correção de erros direta, 244-245
 bit de redundância, 244-246
 bit, 244-245
 dados retransmitidos, 244-245
 erros simples, 244-245
 fórmula para o bit de redundância, 245-246
 múltiplos bits, 248
 rajadas, 248
Correio eletrônico; *veja e-mail*
CPI, 411-413
CR, 749
CRC, 235-236, 238-240, 246-248
 ATM, 408-409
 base, 238-239
 bit de redundância, 238-239
 divisão módulo-2, 239-240
 divisão, 239-240
 divisor, 239-241
 função do receptor, 238-239
 função do transmissor, 238-239
 gerador, 239-240
HDLC, 267
 performance, 242
 polinomial, 240-241
 polinômios padronizados, 240-242
PPP, 278
 receptor, 239-240
 representação do divisor, 248-249
 transmissor, 239-240
 verificador, 240-241
 visão geral, 238-239
CRC-32, 308-309
Ethernet, 308-309
wireless, 335-336
Criptografia com chave privada, 693-694
Criptografia com chave pública, 703-704, 715-716, 723-724
 algoritmo RSA, 704-705
 desvantagens, 703-704
 verificação da chave, 703-704
Criptografia com chave secreta, 704-705
Criptografia com chave simétrica, 693-694
Crosstalk, 176
CS, 409-410
CSMA, 289-292
 colisão, 291-292
 demultiplexador, 301
 multiplexador, 301
 produto interno, 300-301
CSMA/CA, 293-294
 confirmação (ACK), 294-295
wireless LAN, 294-295
CSMA/CD, 292-293, 308
CU, 577-578
Ethernet, 293-294
Ethernet full-duplex, 316-317
 tamanho máximo do frame, 308-309
 transmissão de frame, 308
wireless, 335-336
Cyclical Redundancy Check; *veja CRC*
- D**
- DA, 308-309, 588-589
Dados, 33-34, 73, 253-254
 transmissão, 73
Dados analógicos, 73
Dados autenticados, 735
Dados digitais, 73, 129-130
Dados em tempo real, 675-676
D-AMPS, 372-373
Data Encryption Standard; *veja DES*
Data Link Connection Identifier; *veja DLCI*
Data Over Cable System Interface Specification; *veja DOCSIS*
Datagrama, 427-428, 462-464
Datagrama IP
 cálculo do tamanho do cabeçalho, 464-465
 campo checksum, 465-466
 campo endereço de destino, 465-466
 campo endereço de origem, 465-466
 campo protocolo, 464-465
 campo serviços diferenciados, 464-465
 campo tamanho do cabeçalho, 462-464
 campo tamanho total, 464-465
 campo time-to-live (TTL), 464-465
 campo versão, 462-464
 encapsulamento do segmento, 535-536
 erros, 469-470
 fragmentação, 466
 opções, 467-468
 protocolo de destino, 464-465
 remontagem, 466-467
 saltos permitidos, 464-465
 tamanho, 464-465
Datagrama UDP, 532-533
DCF, 332-333
DCT, 679
 caso com duas escalas de cinza, 679
 caso da escala uniforme de cinza, 679
 caso Gradiente, 680
 valor AC, 679
DDNS, 614-615
DDS, 205-207
 aluguel da linha digital, 206-207
DE PHP, 577-578
Decibel, 92-93
Decifragem, 693-694
Default gateway, 469-470
DEL, 751-752
Demodulador, 156-157
 função, 141-142
Demultiplexação, 156-157, 528-529
 filtros, 156-157
 camada de transporte, 528-529
Demultiplexador (DEMUX), 155
CDMA, 299-300
Dense WDM, 161-162
Densidade, 182-183
Departamento de Defesa; *veja DoD*
DES, 698-699
 tripla, 701
Descriptores do tráfego, 559
Deslocamento de fase, 77-78
Deslocamento em frequência, 133-135
 FSK, 139-140
Desmantelamento da AT&T, 206-207
Destino inalcançável, 468-469
- Detecção de erros*, 233-235
checksum, 242
CRC, 238-239
Ethernet, 308-309
Frame Relay, 307-308
HDL, 267
 teste de paridade, 235-236
DHCP, 441-442
 concessão, 441-442
DDNS, 614-615
 estado de inicialização, 442-443
 estado de religação, 443-444
 estado de renovação, 443-444
 estado de seleção, 442-443
 estado de solicitação, 442-443
 estado ligado, 443-444
 protocolo de configuração dinâmica, 441-442
 transição de estados, 442-443
DHCPACK, 442-444
DHCPDISCOVER, 442-443
DHCPOFFER, 442-443
DHCPREQUEST, 442-443, 443-444
Diagrama de estados finitos,
 cliente, 541-543
 servidor, 542-543
Diagrama de transição de estados, 278, 541-543
Dibit, 136, 137-139
 taxa de modulação (*baud rate*), 139-140
Digital vs. analógico, 73
Digital AMPS; *veja D-AMPS*
Digital Data Service; *veja DDS*
Digital Service Unit; *veja DSU*
Digital Signal Service; *veja DS*
Digital Subscriber Line Access Multiplexer; *veja DSLAM*
Digital Subscriber Line; *veja DSL*
Dígito, 753-754
 mais significativo, 753-754
 menos significativo, 753-754
 ordem do, 753-754
Dinâmico, 483-484, 653-654
Direct Sequence Spread Spectrum; *veja DSSS*
Discagem, 203-205
Discrete Cosine Transform; *veja DCT*
Discrete Multitone Technique; *veja DMT*
Dispositivo de conectividade, 60-61, 340
Distance Vector Multicast Routing Protocol; *veja DVMP*
Distorção, 93-94, 183
Distribuição da chave simétrica, 718-719
 chave, 694-695
Distributed Coordination Function; *veja DCF*
Distributed Inter Frame Space (DFIS), 333-335
Divisor, 217-218
DLCI, 307-308
 Frame Relay, 300-400
DLE, 749
DMT, 214, 216-217
 divisão da largura de banda, 214-215
FDM, 214
QAM, 214
VDSL, 216-217

Hidden page

- Estratégia de persistência, [291-292](#)
 1-persistente, [292-293](#)
 estratégia não persistente, [292-293](#)
 estratégia persistente, [292-293](#)
- Estratégia de transição
 pilha dupla, [475](#)
 tradução do cabeçalho, [475-476](#)
 tunelamento, [475](#)
- Estrela, 39-40
- Etag, [411-412](#)
- Etapas envolvidas, 460-461
- ETB, 750
- Ethernet
 ACK, 308
 AUI, [310-311](#)
 BNC, 179-180
bridged, [313-314](#)
 campo de dados (*payload*), 308-309
 campo tamanho/tipo, 308-309
 campos, 308
 capacidade compartilhada, [313-314](#)
 comutada, [314-315](#)
 CRC, 308-309
 CSMA/CD, [293-294](#), 308, [316-317](#)
 domínio de colisão, [314-315](#)
 endereçamento, 309-310
 endereços, 309-310
 exemplo de rede, 435-436
frame MAC, 308
full-duplex, [315-316](#)
 gerações, 307
 implementações em banda base, [312](#)
 MAC, subcamada de controle, [316-317](#)
 MAU, [311-312](#)
 notação hexadecimal, 309-310
 padrão, 308
 préâmbulo, 308-309
 SA, 308-309
 subcamada PLS, [310-311](#)
 tamanho do *frame*, 308-310
Thick, [312-314](#)
Thin, [312-313](#)
 transceptor, [311-312](#)
- Ethernet par trançado; *veja* 10Base-T
- Ethernet sem fios, [329](#)
- ETX, 749
- Extended Service Set; *veja* ESS
- Extensão do cabeçalho, [474](#)
- Extremely High Frequency; *veja* EHF
- F**
- Fall-back*, 142-144
- Fall-forward*, 142-144
- Fase, 77-78, 129-130, 145, 149-150
 AM, 146-147
 ASK, 131
 definição, 77-78
 e valor do *bit*, 135-136
 FM, 147-149
 FSK, 133
 onda senoidal, 77-78
 PM, 149-150
 PSK, 135-136
 QAM, 137-139
- Fase desconexão, [305-306](#)
- Fast Ethernet, 308, [316-317](#)
 auto-negociação, [317](#)
- codificação, [317](#)
 compatibilidade inversa, [317](#)
- Fator de reuso, 369-370
 GSM, [374-376](#)
 IS-95, [377-378](#)
- FCC, 187-188, [378-379](#)
 endereço, 789
- FCS, [267](#)
- FDM, 155-156
 analogia, 155-156
 aplicações, 160
 bandas de segurança, 155-156
 canais, 155-156
 conceito, 155-156
 FFMDS, [297-298](#)
 implementação, 160
 OFDM, [332-333](#)
 portadora, 155-156
 processo, 156-157
 quando usar, 155-156
 telefonia celular, 160
 TV, 160
- FDMA, [296-298](#), 371-375, [378-379](#)
- FECD, [309-400](#)
 receptor, [567-568](#)
- Federal Communications Committee; *veja* FCC
- FE, 749
- FHSS, [330-331](#)
bluetooth, [340-341](#)
 função, [330-331](#)
- Fiber Link Ethernet; *veja* 10Base-FL
- Fibra, 161, 220-221
- Fibra óptica, 180-183
 aplicações, 184-185
 atenuação, 184-185
 ATM, [401](#)
 capacidade da banda, 220-221
 casca, 183
 composição, 183
 conectores, 183
 conectividade, 184-185
 custo, 184-185
 densidade, 182-183
 desvantagens, 184-185
 HFC, 217-218
 instalação/manutenção, 184-185
 jaqueta externa, 183
 Kevlar, 183
 LAN, 184-185
 largura de banda, 184-185
 luz, 180-181
 materiais corrosivos, 184-185
 modos de propagação, 181-182
 monomodo, 181-183
 multimodo, 181-182
 núcleo, 183
 padronização, 220-221
performance, 183-184
 peso do cabo, 184-185
 propagação unidirecional, 184-185
 reflexão, 181-182
 ruído eletromagnético, 184-185
 tamanhos, 183
 TV a cabo, 184-185
 vantagens, 184-185
- WDM, 161
- Fila, [561-562](#)
- Fila FIFO, [569](#)
 fila de prioridade, [569-570](#)
leaky bucket, [571-572](#)
- File Transfer Protocol; *veja* FTP
- Filtragem, [352](#)
- Filtro
 ADSL, 214-215
 conexão local, 214
- Filtro fixo, 576
- Final, [268](#)
- FIN-WAIT-1, [542-543](#)
- FIN-WAIT-2, [542-543](#)
- Firewall, 740-741
 filtro de pacotes, 740-741, 749
proxy, [741-742](#)
- Física, [308-309](#)
- Flickering, [666-667](#)
- Fluxo de entrega, [535](#)
- Fluxo em rajadas, [560-561](#)
- Fluxograma
ALOHA, [290-291](#)
 LAN sem fios, [333-335](#)
- FM, 145, 147-149
 espectro, 147-149
 estação, 147-149
 largura de banda, 147-150
- Footprint, [380-381](#)
 LEO, [384](#)
- Formato do endereço, [568-580](#)
- Fórmula de Shannon, 143-144
- Fórum, 47-49
- Fórum ATM, [401](#)
 endereço, 789
- Forward Explicit Congestion Notification; *veja* FECN
- FQDN, [604-605](#)
 servidor DNS, [604-605](#)
- FRAD, [399-400](#)
- Fragmentação, 420-421, 466, [474](#)
bit mais fragmentos, 466-467
bit não fragmentar, 466-467
 campo de identificação, 466-467
campo flag, 466-467
campo offset, 466-467
 campos no cabeçalho, 466-467
 definição, 466
 mensagem de erro ICMP, 466-467
offset, 467-468
 remontagem, 466-467
 sem fios, [335-336](#)
- Frame, 59, 161-162, [267](#), [397-400](#), 566-[569](#)
bluetooth, [342-343](#)
 HDLC, [266-267](#)
 MPEG, [667-668](#)
 TDM, 161-163
 vídeo, [666](#)
- Frame 1-slot, [342-343](#)
- Frame bidirecional, [681-682](#)
- Frame Check Sequence; *veja* FCS
- Frame danificado, [254-255](#)
- Frame de controle, [335-336](#)
- Frame de dados, [336-337](#)
poll, [295-296](#)
- Frame de gerenciamento, [335-336](#)
- Frame de supervisão; *veja* S-frame

Hidden page

- HEC, [410-411](#)
 HF, 186-187
 HFC, 217-219
 Hierarquia
 servidor de nomes, [606-607](#)
 sub-rede, 437-438
 Hierarquia analógica, sistema telefônico, 157-159
 High bit rate Digital Subscriber Line; *veja* HDSL
 High level Data Link Control; *veja* HDLC
 High Rate DSSS; *veja* HR-DSSS
 Hipermídia, [648-649](#)
 Hipertexto, [648-649](#)
 Homepage, [648-649](#)
Hop-to-hop, 59
Host
 consulta ARQ, [458-459](#)
 tabela de roteamento, 469-470
Host de destino, remontagem, 466-467
Hostid, 431-432, 436, 438-439
 HR-DSSS, [332-333](#)
 HTML, [649-651](#)
 HTTP, [641-643](#), [649](#), [682-685](#)
 cabecalho geral, 645-646
 cabecalho MIME, [641-642](#)
 cabecalho pedido, 645-646
 cabecalho resposta, 645-646
 cabecalho, 645
 campo de *status* do código, [643-644](#)
 campo versão, [642-644](#)
 cliente, [642](#)
 comandos inseridos, [641-642](#)
 exemplo de recuperação de informação, [646-647](#)
 exemplo de transmissão de dados, 647-648
 formato da mensagem, [641-642](#)
 formato do cabecalho, 645-646
 linha de *status* da frase, [643-644](#)
 mensagem resposta, [643-644](#)
 métodos, [642-643](#)
 porta, 787
 servidor, [642](#)
 similaridade com FTP, [641-642](#)
 similaridade com o SMTP, [641-642](#)
 URL, [642-643](#)
 WWW, [641-642](#)
Hub (concentrador), 217-218
Hub, 40, [351](#)
 HyperText Markup Language; *veja* HTML
 Hypertext Transfer Protocol; *veja* HTTP
- I**
- IAB, 790
 IANA, 527
 ICANN, 790
 ICMP, 467-468
 controle de erros, 468-469
 correção de erros, 468-469
 encapsulamento, 467-468
 mensagem consulta, 469-470
 mensagem de erro, 468-469
 mensagem de redirecionamento, 469-470
 mensagem destino inalcançável, 470-471
 mensagem problema nos parâmetros, 469-470
 mensagem *source quench*, 468-469
 mensagem *tempo excedido*, 469-470
 mensagens da máscara de endereços, 470-471
 mensagens *echo/reply*, 470-471
 mensagens timestamp, 470-471
 pedido/aviso roteador, 470-471
 propósito, 468-469
 relatório de erros, 468-469
 tipos de mensagens, 468-469
 ICMPv6, [474](#)
 ID de usuários, [412-413](#)
 Identificação da multiplexação, [411-412](#)
 Identificação de falhas, 39-40
 Identificação de grupo, 500-501
 Identificador de caminho virtual (VPI), [405-406](#); *veja também* VPI
 Identificador de canal, [410-411](#)
 Identificador de circuito virtual, 392; *veja também* VCI
 IEEE, 47-48
 endereço, 789
 Projeto 802, 230-231
 IEEE 802.11, 329
 IEEE 802.15, [330-340](#)
 IESG, 790
 IETF, [680-690](#), 790
 IFG, [294-295](#)
 Iframe, [681-682](#), 671-672
 IGMP, 498-499
 campo cabeçalho, 500-501
 campo endereço de grupo, 500-501
 campo tipo, 500-501
 campo tipo de resposta máxima, 500-501
 consulta ao roteador, 499-500
 consulta de permanência no grupo, 502
 entrando em um grupo, 500-502
 função, 499-500
 host membro, 500-502
 lista de hosts, 500-502
 lista de roteadores, 503-504
 membro leal, 500-501
 mensagem de consulta, 499-500
 monitoramento dos membros do grupo, 502
 relatório de membros, 499-500
 relatório de saída, 500-502
 resposta atrasada, 502-503
 roteador de distribuição, 500-502
 roteador membro, 500-502
 ILEC, 203, 207-208
 POP, 203
 Imagem, 35
 IMAP4, 629-631
 IMP, 44-45
 Implementação, [318](#)
 MDI, [318](#)
 MII, [318](#)
 subcamada de reconciliação, [317](#)
 subcamada MAC, [317](#)
 subcamada PHY, [318](#)
 IMT-DS, [379-380](#)
 IMT-FT, [379-380](#)
 IMT-SC, [379-380](#)
 IMT-TC, [379-380](#)
- Incumbent Local Exchange Carrier; *veja* ILEC
 Indicador de tamanho, [410-413](#)
 Índice de refração, 181-182
 Informação, 35, 73
 Infra-estrutura de chave pública (PKI), [725-726](#)
 Infra-estrutura de rede, [329-330](#)
 Infrared Data Association (IrDA), 189
 Initialization Sequence Number; *veja* ISN
 Institute of Electrical and Electronic Engineers; *veja* IEEE
 Integridade, 721-723
 assinatura digital, [712-713](#)
 protocolo AH, [734-735](#)
 Inter Frame Space (IFS), [333-335](#)
 Intercalando
 montando o *frame*, 162-163
 rede de células, [402-403](#)
 TDM síncrono, 162-163
 TDM, 162-163
 Interconectividade, 47-48
 Interexchange Carriers; *veja* IXC
 Interface, modelo da Internet, 56
 Interface Mensagem Processor; *veja* IMP
 Interface socket
 cliente, [507-509](#)
 do servidor, [506](#)
 servidor concorrente orientado à conexão, [506-507](#)
 servidor iterativo sem conexão, [506](#)
 Interferência, 176
 LAN, 177-179
 malha local, 202-203
 performance, 177-179
 rede telefônica, 177-179
 RJ45, 177-179
 Interim Standard-95; *veja* IS-95
 International Organization for Standardization; *veja* ISO
 International Telecommunications Union; *veja* ITU
 International Telecommunications Union-Telecommunication; *veja* ITU-T
 Internet, 43-44, 44-45
 abordagem de datagrama, 427-428
 comunicação, [501](#)
 conceito, [458](#)
 corrente, 44-45
 definição, 43-44, 483-484
 DNS, [608-609](#)
 endereço físico, [458](#)
 endereço lógico, [458](#)
 entrega de pacotes, [458](#)
 exemplo, 435-436
 história, 43-44
 pacote, [458](#), 484
 padrão, 48-49
 programas aplicativos, [501](#)
 recursos requeridos por cada computador, 441-442
 rede de comutação de pacotes, 425-426
 representação gráfica do roteamento
 link state, 490-491
 resumo, 48-49
 Internet Control Message Protocol; *veja* ICMP

Hidden page

- sinal de áudio estéreo, 149-150
 sinal de áudio, 146-149
 sob demanda, 168-169
 super-grupo, 159
 taxa de transmissão, 86-87
 telefonia celular, 160
 tráfego em tempo real, 687-688
LATA, 202-203, 206-208
 comunicação, 203
 POP, 203-204
LCP, 278-280
Leaky bucket, 570-571
token bucket, 572-573
LEC, 202-203, 206-207, 777-778
LECS, 778-779
 Lei de Kepler, 380-381
LES, 778-779
LF, 186-187, 749
LI, 412-413
 Limitação do tipo de serviços, 576-577
Linha T, 166-167
 bit de sincronização, 166-167
 capacidade, 167-168
frame, 166-167
overhead, 166-167
 taxa de acesso, 578-579
 taxa de dados, 167-168, 578-579
Linha telefônica, 139-141
 capacidade, 90-91
 largura de banda, 139-141
Linhas E, 167-168
Linhas T, 166-167
 linhas E, 167-168
 multiplexação, 166-167
 PCM, 116
 rajada de dados, 396-397
 relação com os serviços DS, 166-167
 tamanho do *frame*, 166-167
 transmissão analógica, 166-167
Link, 38-39, 155-156, 280-281
stub, 490-491
 transitório, 490-491
 virtual, 490-491
Link Control Protocol; veja LCP
Link ponto a ponto, 490-491
Link State Advertisement; veja LSA
LLC, 230-231, 308
LMI, 400
Local Address Transport Area; veja LATA
Local Area Network; veja LAN
Local Exchange Carrier; veja LEC
Local Management Information; veja LMI
Locator, 642-643
LRC, cálculo, 237-239
LSA, 491-492
 base de dados de *link state*, 493-494
link com a rede, 492-493
link com o roteador, 492-493
link com o roteador de borda, 492-493
link externo, 493-494
 rede, 492-493
LSCAP, 343-344
Luz, 175, 176
 Luz infravermelha, 175
 Luz ultravioleta, 175
 Luz visível, 175
- M**
MA, 274-275, 289-290
MAC, 230-231, 308
 protocolos específicos, 230-231
Mail exchanger, 620
Malha, 39-40
MAN, 42-43
 administração, 42-43
 LANs, 42-43
 uso da, 42-43
Manchester diferencial, 107-110
Mapeamento
 dinâmico, 458
 estático, 458
host_file, 603-604
 nome em endereço IP, 611-612
Máquina de estados finitos, exemplo, 541-543
Marcador, 577-578
Markup Language, 650-651
Máscara, 437-438
 endereçamento baseado em classes, 440-441
Máscara de sub-rede, 438-440
Máscara padrão, 438-439
Mascaramento em frequência, 677-678
Mascaramento temporal, 667-668
Matricial, 772-773
 comutador multiestágios, 196-197
Matriz de Walsh, 299-300
MAU, 311-312
Maximum Transmission Unit (MTU), 466
MBONE, 505-506
MD5, 725-726 (714-715)
MDI, 312, 318
 Gigabit Ethernet, 322
Media Player, 682-683
Medium Attachment Unit; veja MAU
Medium Dependent Interface; veja MDI
Medium Independent Interface; veja MII
Meio, 33, 34
 banda larga, 85-86
 banda limitada, 85-86
Meio de transmissão, 69-70, 73, 175
 camada física, 69-70
 posição, 175
 sinal composto, 81-82
 tipos, 175
Meios
 ar, 184-185
 guia de onda, 173, 176
Memória de acesso aleatório; veja RAM
Mensagem, 34
Mensagem BYE, 600-601
Mensagem CANCEL, 681-682
Mensagem consulta (query), 469-470, 499-500
 especial, 502
 ICMP, 468-469
 tempo de resposta, 502
Mensagem de atualização, 497-498
Mensagem de consulta especial, 502
Mensagem de consulta geral, 502
Mensagem DNS
 cabeçalho, 613-614
 campo de registro de informação, 614-615
- campo de registro de resposta, 613-614
 campo *flags*, 613-614
 campo identificação, 613-614
 campo registro de autoridade, 614-615
 campo registro de pedido, 613-614
 seção autoridade, 614-615
 seção de informação adicional, 614-615
 seção resposta, 614-615
Mensagem GET, 672-675
Mensagem INVITE, 600-601 (681-682)
Mensagem keepalive, 497-498
Mensagem notification, 497-498
Mensagem OPEN, 497-498
Mensagem OPTIONS, 681-682
Mensagem PATH, 575
Mensagem PLAY, 673-675
Mensagem problema nos parâmetros, 469-470
Mensagem REGISTER, 681-682
Mensagem relatório de erros ICMP, 468-469
Mensagem Resv, 575
Mensagem SETUP, 673-675
Mensagem solicitação/anúncio do roteador, 470-471
Mensagem source-quench, 468-469
Mensagem TEARDOWN, 673-675
Mensagem tempo excedido, 469-470
Mensagens da máscara de endereços, 470-471
Mensagens echo/reply, 470-471
Mensagens timestamp
 relógio de sincronização, 470-471
round-Trip Time, 470-471
Metafile, 673
Método additive increase, 565-567
Método de acesso, 229-230, 308
Método de Diffie-Hellman, 718-719
Métrica, 484, 488-489
 OSPF, 488-489
 tipo de serviços, 488-489
TOS, 484
Metropolitan Area Network; veja MAN
MF, 186-187
Microchave, 772-773
Microchaves, 186-189
 antena corneta, 187-189
 antena prato parabólico, 187-189
 antena unidirecional, 187-189
 aplicações, 189
 banda, 187-189
 frequências, 187-189
 porta IrDA, 189-190
 propagação, 187-189
 unidirecional, 187-189
MID, 411-412
MII, 318
AUI, 318
 características, 318
 Fast Ethernet, 318
 recepção de dados, 318
MIME
 alternative, 623-624
 basic audio, 625
 cabeçalho *content-description*, 626-627
 cabeçalho *content-id*, 626-627
 cabeçalho *content-transfer-encoding*, 625

Hidden page

Notação hexadecimal com dois pontos, 471-472
 NRM, [265-266](#)
 NRZ, 107-108, 323-324
 tipos, 107-108
[NRZ-L](#), 107-109
[NRZ-L](#), 107-109
 NSP, 45-46
 NUL, 760
 Número binário, 754
 Número de bytes, 536-537
 Número de confirmação, 529-530, 536-538
 Número de porta, [526-527](#)
 conhecido, 527
 efêmero, [526-527](#), [593-594](#)
 lista, 534-535
 processo, 527
 TCP, 534-535
 UDP, 532-533
 Número de repetições, [633-634](#)
 Número de sequência, [258-259](#), [409-412](#), 529-531, 536-538, 687
 Nyquist
 canal sem ruído, 89-90
 exemplo, 117-118
 taxa de transmissão, 89-90

O

Octeto, 428-429
 OFDM, [332-333](#)
 Onda quadrada, 81-82
 Onda senoidal, 75
 características, 77-78, 129-130
 descrição, 75
 frequência, 75-76
 período, 75-76
 Ondas de infravermelho, 186-187, 189
 Ondas de rádio, 175, 186-187
 aplicações, 187-189
 banda, 187-188
 frequências, 186-187
 omnidirecional, 187-188
 penetração, 187-188
 propagação ionosférica, 185-186
 rádio AM, 187-188
 OOK (On-Off-Keying), 131-132
 Opções, [280-281](#)
 função, 466
 Open Shortest Path First; *veja* OSPF
 Open System Interconnection; *veja* OSI
 Operação AND, 438-439
 Operação push, [552-553](#)
 Operadora de longa distância, 203
 Órbita, [379-380](#)
 Órbita geoestacionária, [381-382](#)
 Organizações de padronização, 47-48
 Orientado à conexão, conceito, 427-428
 Orthogonal Frequency Division Multiple-
 xing; *veja* OFDM
 OSI, 760
 interoperabilidade, 55-56
 QAM, 138-139
 OSPF, 484-485, 487-488
link ponto a ponto, 490-491
link transitório, 490-491
link virtual, 490-491

pacote hello, 497-498
 rede stub, 490-491
 roteamento baseado no vetor de cami-
 nhos, 496
 roteamento link state, 489-490

P

Packer socket, 595
 Pacote, 425-428, 484
 formato, 425-426
 informação de controle, 425-426
 tamanho, 425-426
 Pacote de configuração, [280-281](#)
 Pacote de resposta, [458-459](#)
 Pacote de tamanho variável, *leaky bucket*, [571-572](#)
 Pacote hello, 497-498
 Pacote ICMP, módulo de saída, 476-477
 Pacote IPv6, [473](#)
 Pacote source-quench, [564-565](#)
 Padrão, 46-47
 comitês de criação, 47-48
 definição, 46-47
 Internet, 48-49
 necessidade de, 46-47
 ratificação, 48-49
 tipos, 47-48
 Padrão *de facto*, 47-48
 Padrão *de jure*, 47-48
 Página, [648-649](#)
 Página da Web, [649-651](#)
Paging, 370-371, 376-377
 PAM, 115-116
 PAP, [281-282](#)
 Par trançado, 176
 aplicações, 177-179
 blindado, 176-177
 componentes, 176
 DSL, 177-179, 216-217
 HDSL, 215-216
 tipos, 176-177
 Para DS, [336-337](#)
 Paradigma cliente-servidor, [526-527](#)
 Paridade ímpar, 236-237
 Paridade par, 235-236
Parked state, [339-340](#)
 Partially Qualified Domain Name; *veja* PQDN
 Passa-baixas, 88-89
 Passa-faixa, 88-89
Passagem do token, [295-296](#)
 Passive open, [592-593](#)
 Password Authentication Protocol; *veja* PAP
 Pbox, 698
 PCM, 115-117
 bits por segundo, 117-118
 codificação binária, 116
 codificação de linha, 116
 PAM, 115-116
 processos, 116
 quantização, 116
 taxa de bits, 118-119
 PCS, [378-379](#)
 Pedido de configuração, [304-305](#)
 Per Hop Behavior; *veja* PHB
 Perdas na compressão, 669-670

Perfis do tráfego, [560-561](#)
 Performance, 37-38, 242
checksum, 243-244
 da rede, [561-562](#)
 Período, 73-76, 380-381
 inverso, 75-76
 unidade, 75-76
 Período de um satélite, [380-381](#)
 Personal Area Network (PAN), [339-340](#)
 Personal Communication System (PCS), [378-379](#)
Pyframe, 671-672
 PGP, 739-740
 Phase Shift Keying; *veja* PSK
 PHB, [577-578](#)
 Piconet, [339-340](#)
Piggybacking, [257-258](#), [268](#), [270-271](#)
 Pilha dupla de protocolos, [475](#)
 PIM, [511-512](#)
 PIM-DM, [511-512](#)
 PIM-SM, [511-512](#)
Pipelining, [265-266](#)
Pixel, 35, [666-667](#), 670-671
 PM, 145, 149-150
 Polaridade, 106-107
 Polinomial, 240-242
 CRC, 240-241, 248-249
 propriedades, 240-241
 representação binária, 240-241
 Política de admissão, [564](#)
 Política de descarte de pacotes, [563-564](#)
 Política de janela, [564](#)
 Política de retransmissão, [564](#)
 Política de roteamento, 496-497
Poll, [268](#), [295](#)
Polling, [268](#), [295](#)
 Ponto a ponto, 37-40
 definição, 38-39
 exemplo de rede, 435-436
 malha, 39-40
 Ponto de acesso; *veja* AP
 Ponto de cruzamento, 196-199
 Ponto de presença; *veja* POP
 Pontos de sincronização, 772
 POP, 203-204
 POP3, 629-631
 Porta
 conhecida, 527
 efêmera, 527
 registrada, 527
 Porta conhecida, 527, [593-594](#), [596](#)
 servidor, [593-594](#)
 UDP e TCP, 787
 Porta de bloqueio (*blocking*), [356](#)
 Porta de encaminhamento, [356](#)
 Porta de entrada, 771
 Porta designada, [356](#)
 Porta dinâmica, 527
 Porta IrDA, 189
 Porta raiz, [355-356](#)
 Porta UDP RTCP, [680-681](#)
 Portadora, 131
 AM, 146-147
 FM, 147-149
 Inter-LATA, 203
 PM, 149-150
 Portadora consum, 202-203
 antes de 1934, 206-207

- após 1996, 206-207
entre 1934 e 1996, 206-207
história, 206-207
Portas de saída, 772
Post Office Protocol, versão 3; *veja POP3*
Potência, 175
satélite, 381
POTS, 201-202
p-persistente, 292-293
PPP, 277-278, 283-284
autenticação, 281-282
authenticating state (autenticação), 278-279
campo controle, 278
campo dados, 278
campo endereço, 278
campo flag, 277-278
campo protocolo, 280-281
establishing state (estabelecimento), 278
FCS, 278
frame, 277-278
HDLC, 277-278
idle state (ocioso), 278
ISP, 277-278
NCP, 283-284
negociação das opções, 280-281
networking state (encapsulamento/transmissão), 278-279
pacote de término do link, 280-281
pilha, 278-279
terminating state (desconexão), 278-279
teste loopback, 280-281
transição de estados, 278
PQDN, 604-605
sufixo, 605-606
Preâmbulo, 308-309
Pretty Good Privacy; *veja PGP*
Prevenção de loops, 496-497
Privacidade, 711-712
protocolo AH, 734-735
Probe, 551-552
Problema n°, 718-719
Processador da camada física, 772
Processador de enlace, 772
Processador de roteamento, 772-773
Processamento distribuído, 37-38
Processo, 503-504
Processo de bootstrap, 441-442
Processos peer-to-peer, 56
Produto banda × atraso de propagação, 264-265
Produto interno, 300-301
Programa, processo, 503-504
Programa cliente
atividade, 591-592
camada de transporte, 522-523
número de porta, 526-527
Programa servidor, 527
atividade, 591-592
número de porta, 527
Programas aplicativos, 591
Programas UDP, 780
Propagação direcionada, 185-186, 381-382
antena de microondas, 185-189
Propagação ionosférica, 185-186
Propagação superficial, 185-186
Proteção do número de seqüência, 409-410
Protocol Independent Multicast, Dense Mode; *veja PIM-DM*
Protocol Independent Multicast, Sparse Mode; *veja PIM-SM*
Protocol Independent Multicast; *veja PIM*
Protocolo, 34, 46-47, 56
Protocolo AH, 734-737
Protocolo Authentication Header; *veja Protocolo AH*
Protocolo de acesso à caixa de correio, 629-630
Protocolo de handshake, 738-739
Protocolo de Otway-Rees, 722-723
Protocolo de roteamento, 420-421, 483-484
Protocolo de troca de dados, 737-738
Protocolo do endereço de destino, 459-460
Protocolo orientado à conexão, 777-778
Protocolo Ponto a Ponto; *veja PPP*
Protocolo pull, 629-630
Protocolo push, 629-630
Pruning (poda), 507-508
Psicoacústica, 677-678
PSK, 129-130, 135-137, 141-142
ASK, 135-137, 141-142
binário, 135-136
exemplo de largura de banda, 136-139
FSK, 135-136
largura de banda, 136-137
limitações, 137-139
modem, 141-142
taxa de bits, 136-137
Pulse Amplitude Modulation; *veja PAM*
Pulse Code Modulation; *veja PCM*
PVC, 392-394, 405-406
ATM, 405-406
estabelecimento, 405-406
- Q**
- Q.931, 684-685
QAM, 130, 137-139, 214
ASK, 138-139
codificação treliça (trellis), 142-144
largura de banda, 138-139
seleção de erro, 142-144
variações, 137-139
QoS, 344-345, 568-569
ATM, 580-581
bluetooth, 344-345
como melhorar, 569
condicionamento do tráfego, 570-571
controle de admissão, 573-574
DF, 576-577
Frame Relay, 578-579
IntServ, 573-574
leaky bucket, 570-571
rede comutada, 578-579
reserva de recursos, 573-574
Q-PSK, 136
Quadrature Amplitude Modulation; *veja QAM*
Qualidade de serviços, 401; *veja também QoS*
Quantização, 669-670
PCM, 115-116
Queda da linha de transmissão, 40
Queda do link, 381-382
- R**
- Rádio AM, 160
Rádio FM, 147-149, 160
Radio Government; *veja RG*
Rádio na Internet, 666
Raio cósmico, 175
Raio gama, 175
Raio X, 175
Rajada de dados, 396-397
controle de tráfego, 580-581
Frame Relay, 396-397
linhas T, 396-397
Rajada de erros, 233-235
RAM, 199-200
RARF, 458
Raw socket, 596
Razão sinal-ruído; *veja SNR*
RBOC, 206-207
RCH, 217-218
Realm, 728
Real-Time Streaming Protocol (RTSP), 673-675
Real-Time Transport Protocol (RTP), 534-535; *veja também RTP*
Receive Not Ready; *veja RNR*
Receptor, 34
controle de fluxo, 230-231, 253-254
reserva, 575
Rede, 37-38, 582
confiabilidade, 37-38
critérios, 37-38
definição, 43-44
híbrida, 743-745
performance, 37-38
privada, 742-743
tipos, 42
Rede corporativa, 43-44
Rede de células, 402-403
conceito, 402-403
exemplo, 402-403
multiplexação, 402-403
stream, 402-403
transmissão em tempo real, 402-403
VC, 404-405
vs rede de frames, 402-403
Rede de frames, 401
Rede de satélites, 379-380
Rede híbrida cabo coaxial/fibra óptica; *veja HFC*
Rede stub, 490-491
Rede telefônica, 201-202
componentes, 201-202
comutador, 204-205
largura de banda, 204-205
serviço analógico alugado, 205-206
serviço analógico comutado, 204-205
serviço digital, 205-206
serviços analógicos, 204-205
Redes backbones, 357-358
Redirecionamento de mensagem, 469-470
Redundância, 235, 678-679
checksum, 242

- codificação *trellis*, 142-144
 conceito, 235
 CRC, 238-239
 dados duplicados, 235
 Reflexão, 235-236
 Refração, 180-181
 Regional Bell Operating System; *veja* RBOC
 Registration/Administration/Status (RAS),
[683-684](#)
 REJ, [268-269](#)
 Reject; *veja* REJ
 Relatório de membros, 499-502
 Relatório de saídas, 499-502
 Relógio (*timer*) de persistência, 551-552
 Relógio de retransmissão, [550-551](#)
 Relógio *keepalive*, [552-553](#)
 Relógio *time-wait*, propósito, [552-553](#)
 Repetidor, [349](#)
 amplificador, [349](#)
 anel, 40-41
 HDSL, 215-216
 hub, [351](#)
 localização, [351](#)
 segmento, [350](#)
 Representação no domínio da frequência, 78-79
 Representação no domínio do tempo, 77-78
 Request for Comment; *veja* RFC
 Request to Send (RTS), [333-335](#)
 Reserva, [295](#), 573-574
 renovação, 576-577
 Reserva de *frame*, [295](#)
 Resetando a conexão, [540-541](#)
 Resolução de endereços IP – nomes, 611-612
 Resolução iterativa, [612-613](#)
 Resolução nome-endereço, [611](#)
 Resolução recursiva, [611-612](#)
 Resposta, DNS, [613-614](#)
 Resposta *unicast*, [458-459](#)
 Retransmissão, 565-566
 Return to Zero; *veja* RZ
 Reverse Address Resolution Protocol; *veja* RARP
 Reverse Path Broadcasting; *veja* RPB
 Reverse Path Forwarding; *veja* RPF
 Reverse Path Multicasting; *veja* RPM
 Revestimento, 178-179
 RFC, 48-49, 785
 lista, 785
 níveis de maturidade, 48-49
 RG, 179-180
 cabô coaxial, 179-180
 taxas de transmissão, 179-180
 RIP, 484-486, 532-534
 algoritmo de atualização, 485-486
 encapsulamento, 487-488
 porta, 788
 RJ45, 177-179
 RNR, [268-269](#)
 roteamento baseado no vetor de distâncias, 496
Routing, 370-371
 Roteador, 60-61, 483-484
 backbone, 488-489
 borda de área, 488-489
 borda de *autonomous system*, 487-488
 componentes, 771
 comutação, 772-773
 designado pai, [507-508](#)
 endereço, 470-471
 estrutura, 771
 fragmentação, 420-421, 466
 multicast; *ver* Roteador *multicast*
 porta de entrada, 771
 porta de saída, 772
 rendezvous, 509-510
 sub-rede, 437-438
 Roteador de borda de AS, 496
 Roteador *multicast*, 500-501
 identificação de grupo (*groupid*), 500-501
 propósito, 502
 Roteamento
 camada de rede, 60-61
 dinâmico, 447-449
 estático, 447-449
 interno e externo, 484
 link state, 489-490
 padrão, 447-449
 para host específico, 447-449
 para rede específica, 447
 próximo salto, 446-447
 sub-rede, 437-438
 vetor de distâncias vs. *link state*, 489-490
 vetor de distâncias, 485-486
 Roteamento baseado em LSA, 489-490
 BGP, 496
 representação gráfica, 490-491
 tabela de roteamento, 494-495
 Roteamento baseado no próximo salto, 446-447
 Roteamento baseado no vetor de caminhos, [405-407](#)
 Roteamento baseado no vetor de distâncias, 485-486
 BGP, 496
 compartilhando com os vizinhos, 485-486
 compartilhando informação, 485-486
 compartilhando intervalos, 485-486
 tabela de roteamento, 500-502
 Round Trip Time; *veja* RTT
 Routing Information Protocol; *veja* RIP
 RPB, [507-508](#)
 RPF, 788
 RPF, 506-507
 RPM, [507-508](#)
 RR, [268-269](#)
 RSpec, 573-574
 RSVP, 573-575
 estilo de reserva, 576
 IntServ, [574-575](#)
 margem de reserva, 576
 mensagem, [575](#)
 RTCP, [680](#)
 RTP, 679
 RTT, [550-551](#)
 algoritmo de Karn, 551-552
 cálculo, [550-551](#)
 como um função dos valores RTTs anteriores, 551-552
 Ruido, 93-94
 ASK, 131-132
 cabô coaxial, 131
 crosstalk, 93-94
 efeito na amplitude, 131-132
 impulsivo, 93-94
 PSK, 135-136
 QAM, 137-139
 serviço digital, 205-206
 térmico, 93-94
 Ruído de quantização, modem, 143-144
 RZ, 107-110
- S**
- SA, 308-309, [588-589](#)
 SAR, [409-410](#)
 Satélite, [379-380](#)
 faixa de frequência, [381-382](#)
 geoestacionário, [381-382](#)
 truncamento (*trunk*), 202-203
 Satélite de órbita média; *veja* satélite MEO
 Satélite GEO, [381-382](#)
 Satélite LEO, [381-384](#)
 Satélite MEO, [381-383](#)
 S-Box, 698-699
 Scatternet, [339-340](#)
 SCO, [342-343](#)
 SDSL, 215-216
 SE, 576-577
 SEAL, [412-413](#)
 Secundário, *polling*, [295-296](#)
 Secure Sockets Layer (SSL), [737-738](#)
 Security Association (SA), [734](#)
 Security Parameter Index (SPI), 735; *veja também* SPI
 Segmentação, L2CAP, [344-345](#)
 Segmentação e remontagem; *veja* SAR
 Segmento, 62-63, [314-315](#), [350](#), [535-536](#), 538-539
 campos do cabeçalho, 538-539
 formato, 538-539
 tamanho, 538-539
 TCP, 538-539
 Segmento de cabô coaxial, 217-218
 Segmento duplicado, [548-549](#)
 Segmento FIN, [540-541](#)
 Segmento perdido, [548-549](#)
 Segmento SYN, [540-541](#)
 Segurança, 37-38
 autenticação, [712-713](#)
 camada de aplicação, 739-740
 camada de rede, [733](#)
 camada de transporte, [737-738](#)
 integridade, [712-713](#)
 Segurança de rede, 37-38
 Segurança na camada IP, [733](#); *veja também* IPSec
 Seleção
 endereçamento, [295](#)
 frame, [295](#)
 polling, [295](#)
 Select, [295](#)
 Selective Reject; *veja* SREJ
 Selective Repeat, [563-564](#)
 bidirecional, [264-265](#)
 operação, [262-263](#)

- tamanho da janela de recepção, 262-
[263](#)
 tamanho da janela de transmissão, 262-
[263](#)
 tamanho da janela, [263-264](#)
Selective Repeat ARQ, [258-259](#)
Semântica, 46-47
Senha, [281-283](#)
Senha de convidado, [635-636](#)
Sequência ortogonal, [299-300](#)
Serviço analógico, 204-205
Serviço analógico comutado, 204-205
Serviço confiável, 536-537
Serviço confiável do nível de transporte,
 531-532
Serviço de chamada grátil, 203
Serviço digital, 205-206
Serviço full-duplex, 536-537
Serviço host-to-host, 419, 423-424, [525](#)
Serviço inter-LATA, 203
Serviço intra-LATA, 202-203
Serviço não confiável da camada de trans-
 porte, 531-532
Serviço node-to-node, 101-102, 525
Serviço orientado à conexão, 529-530,
 536-537
Serviço sem conexão, [528-529](#)
Serviços de diretório, 763-764
Serviços de melhor esforço, 462
Serviços de rede, 427-428
Serviços em linhas analógicas alugadas,
 157-159, 205-206
Serviços integrados; *veja IntServ*
Serviços 0800, 205-206
Serviços origem-destino, 59-60
Serviços 0900, 205-206
Servidor, [526-527](#), [591-593](#)
 - aceitando a chamada, [597-598](#)
 - aceitando o socket, [597-598](#)
 - bind*, [596-597](#)
 - buffer*, [596-597](#)
 - chamada em espera, [596-597](#)
 - clientes, [591-592](#)
 - concorrente orientado à conexão, 593-
[594](#), [592-593](#)
 - concorrente, [592-593](#)
 - definição, [597-593](#)
 - encerrando a comunicação com o *soc-*
ket, [597-598](#)
 - escrita, [597-598](#)
 - etapas de repetição, [597-598](#)
 - fila, [596](#)
 - iniciando um *socket*, [596](#)
 - iterativo sem conexão, [592-593](#), [596](#)
 - iterativo, [592-593](#)
 - leitura, [597-598](#)
 - listen*, [596-597](#)
 - mensagem *passive open*, [592-593](#)
 - porta conhecida, [593-594](#)
 - porta efêmera, [593-594](#)
 - principal, [607-608](#)
 - processamento de dados, [597-598](#)
 - protocolo da camada de transporte,
[592-593](#)
 - raiz, [607-608](#)
 - recebe, [596-597](#)
 - repetição, [597-598](#)- secundário, [607-608](#)
- TCP, [593-594](#), [596-597](#)
- UDP, [592-593](#)
- Servidor de autenticação; *veja AS*
- Servidor de *proxy*, [741-742](#)
- Servidor de registros, [682-683](#)
- Servidor de *streaming*, 673
- Servidor filho, [596-597](#)
- Servidor pai, [596-597](#)
- Servidor primário, [607-608](#)
- Sessão *active close*, [592-593](#)
- Sessão *active open*, [592-593](#)
- Session Initiation Protocol; *veja SIP*
- SFD, 308-309
- Sframe*, [266-267](#)
- SHA-1*, [714-715](#)
- SHE, 186-187
- Shift Keying, 131
- Short Inter Frame Space (SIFS), [333-335](#)
- Simple and Efficient Adaptation Layer; *veja*
SEAL
- Simplex*, 35-36
- Sinais eletromagnéticos, 73
- Sinal, 73
 - amplitude, 75
 - analógico, 73-74, 129-130
 - analógico composto, 73-74
 - analógico e digital, 73
 - analógico vs sinal digital, 88-89
 - aperiódico, 73-74
 - degradação, 73-74
 - não periódico, 73-74
 - periódico, 73-75
 - tipos, 73, 88-92, 94-97
- Sinal composto, 79-80
 - distorção, 93-94
 - meio de transmissão, 82-83
- Sinal de áudio, 676-677
 - digitalização, 676-677
- Sinal digital, 73-74, 84-85
 - características, 73-74
 - largura de banda, 85-86
 - sinal analógico composto, 85-86
 - vs. sinal analógico, 88-89
- Sinal direto, [564-565](#)
- Sinal e módulo, 116
- Sinal explícito, [564-565](#)
- Sinal implícito, [564-565](#)
- Sinal modulante, 131
- Sinal unitário, 130
- Sincronização, 129-130
 - clock*, 106
 - codificação de blocos, 111-112
 - codificação unipolar, 106-107
 - exemplo, 106
 - IS-95, [374-376](#)
 - Manchester, 109-110
 - Manchester Diferencial, 109-110
 - nível de *bytes*, 120-121
 - RZ, 108-110
 - transmissão assíncrona, 120-121
- Síndrome da janela boba, [547](#)
 - algoritmo de Nagle, [547](#), [548](#)
 - atraso da confirmação, [548](#)
 - causa, [547](#)
 - criada pelo receptor, [548](#)
 - criada pelo transmissor, [547](#)
- solução de Clark, [548](#)
- Sintaxe, 46-47
- Síntese, [713-714](#)
 - função de *hash*, [713-714](#)
 - no receptor, [714-715](#)
 - no transmissor, [714-715](#)
 - PGP, 739-740
 - SIP, [680-681](#)
- Sistema aberto, 760
- Sistema binário, 753-754
- Sistema de numeração, 753-754, 756-757
- Sistema decimal, 753-754
 - conversão de, 757-758
 - conversão para, 757-758
 - para binário, 757-758
 - para hexadecimal, 758-759
 - para octal, 758-759
 - peso e valor, 754
 - ícones, 754
- Sistema distribuído, [329-330](#)
- Sistema hexadecimal, 753-756
- Sistema octal, 753-755
- Sistema telefônico
 - hierarquia, 159
 - multiplexação, 157-159
 - serviço analógico comutado, 157-159
- Slow start*, 565-566, 647-648
- SMTP**
 - caixa de correio do usuário, [620](#)
 - cliente, 628-629
 - comandos do cliente, [627-628](#)
 - comandos, [626-628](#)
 - estabelecimento da conexão, [627-628](#)
 - fases da transferência de *e-mail*, 627-
[628](#)
 - HTTP, [641-642](#)
 - limitações, [621-622](#)
 - mail exchangers*, [620](#)
 - nome do domínio, [620](#)
 - parte local do endereço, [620](#)
 - porta, 788
 - primeiro estágio, 628-629
 - respostas, [626-628](#)
 - segundo estágio, 628-629
 - servidor, 628-629
 - sistema de endereços, [694](#)
 - termino da conexão, [627-628](#)
 - transferência da mensagem, [627-628](#)
- SNMP**: porta, 788
- SNR**, 90-91
- Socket**
 - bind*, [596](#)
 - campo família, 594-595
 - campo protocolo, 594-595
 - campo tipo, 594-595
 - datagrama, 595
 - definição, 594-595
 - endereço local, 595
 - endereço remoto, 595
 - estrutura do, 594-595
 - fechando, [596-597](#)
 - passos de repetição, [596-597](#)
 - raw*, [596](#)
 - recebendo, [596-597](#)
 - stream*, 595
 - transmitindo, [596-597](#)

- Soft handoff*, 370-371
IS-95, 377-378
- Soft state*, 576-577
- SOH, 749
- Solução de Clark, 548
- Soma parcial, 780
- SONET, 406-407
ATM, 406-407
video, 677-678
- Source Routing Bridge (SRB), 356-357
- SP, 750
- SPI, 735
- SREJ, 268-269
- ST, 411-412
- Start Frame Delimiter; *veja* SFD
- Stop-and-Wait ARQ*, 254-257
bidirecional, 257-258
frame perdido, 255
numerando ACK, 254-255
piggybacking, 257-258
relógio (*timer*), 255
resposta de confirmação atrasada, 256-257
resposta de confirmação perdida, 255-256
transmissão normal, 255
transmissor, 254-255
- STP, 176-177
- Stream socket*, 595
- Streaming*, 682-683
Streaming de áudio armazenado, 672
Streaming de áudio/vídeo armazenado, 665, 682-683
servidor de *streaming* e RTSP, 673-675
servidor de *streaming*, 673
servidor de web e *metafile*, 673
servidor de web, 672
- Streaming* de áudio/vídeo em tempo real, 665, 674-675
- STX, 749
- Suavizador, 578-579
- Subcamada de convergência; *veja* CS
- Subcamada de reconciliação, 317, 321-322
subcamada PLS, 317
- Subcamada MAC, 230-231, 308
Fast Ethernet, 316-317
função, 308
Gigabit Ethernet, 321-322
LAN sem fios, 332-333
- Subcamada PLS, 310-311
subcamada de reconciliação, 317
- Subcamada, PHY, 318
- Sub-rede
hostid, 437-438
prefixo, 441-442
roteador, 437-438
sem classe de endereçamento, 441-442
- Substituição, 112-113
monoalfabética, 695-696
polialfabética, 696-697, 705-706
- Sufixo, 605-606
- Sufixo nulo, 605-606
- Supergrupo, 439-440
- SVC, 424-425, 405-406
ATM, 405-406
- Switch* (comutador), 60-61, 195-196, 315-316, 772-773
- backbone* estrela, 350-360
- Banyan, 772-773
- Batcher-Banyan, 773-774
- camada 2, 357-358
- camada 3, 315-316
- exemplo *time-space-time* (TST), 201-202
- matricial, 196-197, 772-773
- rede telefônica, 202-203
- sistema telefônico, 202-203
- tabela, 392-394
- Switch Banyan, 772-773
- Switched Ethernet, 314-315
- Switched Multi-Megabit Data Services (SMDS), 42-43
- Switched*/56, 205-206
assinante, 205-206
circuito, 195-196
comutação, 195-196
divisão de espaço *vs.* divisão de tempo, 200-201
largura de banda sob demanda, 206-207
métodos, 195-196
modem, 206-207
multiestágio, 196-197
necessidade de, 195-196
nós de, 195-196
realimentação, 196
taxa de dados, 205-206
TDM bus, 200-201
usos, 206-207
- Symmetric Digital Subscriber Line; *veja* SD-SL
- T**
- Tabela, comutação de circuitos virtuais, 392-394
- Tabela de roteamento, 446-447, 483-484, 494-495
árvore de caminho mais curto, 494-495
atualização da, 469-470, 487-488
campo próximo salto, 487-488
dinâmico, 448-449, 483-484
entradas, 485-486
estático, 448-449, 483-484
initialização, 487-488
link state, 489-490
- Tabela *lookup*, 682-683
- Tag, 650-651
atributos, 650-651
comum, 651-652
formato, 650-651
- Tamanho da janela, base, 565-566
- Tamanho da rajada comprometido; *veja* B_c
- Tamanho do *frame*, 308-309
- Tamanho excedente da rajada; *veja* B_c
- Tamanho máximo da rajada, 560-561
- Tap, 40, 217-218
- Taxa de acesso, 578-579
T-L, 578-579
- Taxa de amostragem, 117, 676-677
PAM, 117
PCM, 117
teorema de Nyquist, 117
- Taxa de informação comprometida; *veja* CIR
- Taxa de modulação (baud rate), 130, 133-137
- ASK, 136-137
- bit*, 139-140
- exemplo, 131, 139-141
- FSK, 139-140
- PSK, 136-137
- vs* taxa de transmissão, 131
- Taxa de pulsos, 104-105
- Taxa de transmissão, 305, 84-85, 104-105, 130
bit, 139-140
exemplo, 104-105, 131, 139-141
largura de banda, 86-87
vs taxa de modulação, 131
- Taxa máxima de dados, 560-561
- Taxa média de dados, 559
- TCP, 44-45, 462, 525, 531-532, 534-535
arquitetura cliente-servidor, 592-593
bit push, 553
buffer circular, 535-536
controle de erros, 548
controle de fluxo, 543-544
dados, 557-558
divisão, 44-45
DNS, 615-616
e IP, 462
fluxo de entrega, 535
janela deslizante, 543-544
modo *full-duplex*, 540
número ACK, 537-538
número de porta, 534-535
número de seqüência, 537-538
operação push, 552-553
orientado à conexão, 539-540
portas, 788
- programa cliente, 783-784
programa servidor, 795-820
protocolo orientado a *streaming*, 553
relógio (*timer*) de persistência, 549-550
relógios (*timers*), 549-550
resetando, 540-541
segmentação, 633-634
segmento, 535-536, 538-539
segmento duplicado, 548-549
segmento fora de ordem, 549-550
serviço confiável, 536-537
serviço orientado à conexão, 536-537
SIP, 680-681
streaming de áudio/vídeo em tempo real, 674-675
tráfego em tempo real,
urgência de dados, 553
- TCP/IP
camada de aplicação, 763-764
formato do datagrama, 462-464
versão 5, 66-67
- TDD, 340-341
- TDD-TDMA, 340-341
- TDM, 155-156, 161-162
aplicações, 168-169
bits de enchimento (*padding*), 164-165
frame, 161-162
framing, 164
taxa de dados, 161-162
- TDMA, 297-298
- time slot*, 161-162

- TDM assíncrono, ATM, [402-403](#)
 TDM bus, 200-201
 TDMA, [296-298, 374-376, 378-379](#)
bluetooth, [340-341](#)
 Técnica de descoberta da MTU do caminho, [474](#)
 Técnicas de roteamento, 446-447
 Telecommunications Act de 1996, 202-203, 206-207
 Telecomunicação, 33-34
 Teledesic, [385-386](#)
 Telefone celular, 160
 Telefonia celular, 369
 fazendo uma chamada, 369-370
handoff, 370-371
 MSC, 370-371
 potência de transmissão, 369
 primeira geração, 370-371
 procura, 369
 raio, 369
 recebendo uma chamada, 370-371
 segunda geração, 371-372
 sinal de consulta, 370-371
 sinal fraco, 370-371
 terceira geração, [378-379](#)
 Telefonia pela internet, 674-675
 TELNET, [642-643, 649](#)
 porta, 788
 Tempo de espera, [330-331](#)
 Tempo de propagação, 94-96
 Tempo de retransmissão, [550-551](#)
 Tempo real
 buffer de reprodução, 677-678
 limiar, 677-678
 Temporização, 46-47
 Teorema de Nyquist, 117, 676-677
 freqüência, 117
 Teoria da codificação, [297-298](#)
 Terminal escondido, [333-335](#)
 Término da conexão, 530-531, [540-541](#)
 procedimento, 530-531, [540-541](#)
 SMTE, [627-628](#)
 Teste de paridade, 235-236
 bidimensional, 237-238
 quantidade de erros ímpar, 237-238
 quantidade de erros par, 237-238
 simples, 235-236
 Teste de paridade bidimensional, 237-238
 erros não detectáveis, 238-239
 performance, 238-239
 Teste de paridade simples, 235-236
 performance, 237-238
 Teste de redundância, 235-236
 Texto cifrado, 693-694
 RSA, 704-705
 Texto limpo, 693-694
 RSA, 704-705
 TFTP, porta, 788
 TGS, [725-726](#)
 AS, 726-727
 Kerberos, 726-727
 Thick Ethernet; veja 10Base5
 Thicknet; veja 10Base5
 Thin Ethernet; veja 10Base2
Throughput, 94-95, [561-564](#)
Ticker, 720-721
 Ticket-Granting Server; veja TGS
 Time Division Duplexing; veja TDD
 Time Division Multiple Access; veja TDMA
Time slot comutação, 200-201
 Time-Division Multiplexing; veja TDM
Time-out, [548-549](#)
 retransmissão dinâmica, [550-551](#)
Timers (temporizadores/telênicos), 549-550
 de espera, [552-553](#)
Go-Back-N, [259-260](#)
Keepalive, [552-553](#)
 persistente, 551-552
 retransmissão, [550-551](#)
 Time-Slot Interchange; veja TSI
Timestamp, 676-677
 relatório do transmissor, [680](#)
Time-to-live, caching, 612-613
 Tipo de fluxo, 471-472
 Tipo de payload do pacote, [410-411](#)
 Tipo de segmento, [411-412](#)
 Tipos de sistemas telefônicos, 157-159
 TLS, [737-738](#)
Token, [295-296](#)
Token bucket, [570-573](#)
 leaky bucket, [572-573](#)
 medidor, [577-578](#)
Token ring, exemplo de rede, 435-436
Tom dual, 203-204
 Topologia, 38-39
 Topologia barramento, 195-196
 Topologia em malha, 195-196
 Topologia estrela, 195-196
 TP, [403-404](#)
 Trabalhando com sub-redes, 436
 Tradução, 678-679
 camada de apresentação, 761
 Tradução do cabeçalho, [475-476](#)
 Tráfego, tamanho do frame, [401-402](#)
 Tráfego à taxa constante de bits, [560-561](#)
 Tráfego à taxa variável de bits, [560-561](#)
 Tráfego de dados, [550](#)
 Tráfego em rajadas
 leaky bucket, [571-572](#)
 token bucket, [572-573](#)
 Tráfego em tempo real
 controle de erro, 678-679
 mixer, 678-679
 multicasting, 678-679
 número de seqüência, 677-678
 RTP, 679
 TCP, 679
 timestamp, 676-677
 tradução, 678-679
 tradutor, 678-679
Trailer, 57, 59
 Transceptor, [311-312](#)
 1000Base-T, [323-324](#)
 1000Base-X, [322-323](#)
 100Base-FX, [319-320](#)
 100Base-T4, [320-321](#)
 100Base-TX, [318-319](#)
 Fast Ethernet, [318](#)
 Gigabit Ethernet, [322](#)
 Transferência de arquivo, 629-631
 Transferência de dados comutação de circuitos virtuais, 392-394
 Transição, [475](#)
 estratégias, [475](#)
 IPv4 para IPv6, [475](#)
 Transição bidirecional, 490-491
 Transmissão, 73, [403-404](#)
 AMPS, 571-572
 D-AMPS, [377-378](#)
 IS-95
 Transmission Control Protocol; veja TCP
 Transmissão analógica, linhas T, 166-167
 Transmissão assíncrona, 118-122
 Transmissão com perdas, 91-92
 Transmissão em tempo real, 34
 Transmissão não guiada, 175, 184-185
 Transmissão orientada à conexão, [522](#)
 Transmissão paralela, 118-120, 234-235
 Transmissão sem conexão, [522](#)
 Transmissão sem fios, tamanho do frame, [335-336](#)
 Transmissão serial, 118-120
 classes, 118-119
 dispositivo de conversão, 119-120
 rajada de erros, 234-235
 tipos, 119-120
 vantagem, 119-120
 Transmissão síncrona, 118-119, 121-123
 agrupamento de bits, 121-122
 função do receptor, 121-122
 sincronização, 122-123
 vantagem, 122-123
 Transmission Paths; veja TP
 Transmissor, 34, [567-568](#)
 controle de fluxo, [253-254](#)
 Transparência de dados, [271-272](#)
 Transport Layer Security; veja TLS
 Triangulação, [382-383](#)
Tribit, 136-139
 Triplo DES, 701
 Triplo handshake, [540](#)
Trunk, 201-203
 TSI, 198-201
 Tspec, 573-574
 Tunelamento, [475](#), 504-505
 VPN, 745-746
 TV, 160
 TV a cabo, 216-217
 TV pela Internet, [666](#)
 Twisted-pair, 176-177

U

- UA,
- [620-621](#)
-
- UDP,
- [525, 529-533, 679](#)
-
- arquitetura cliente-servidor,
- [592-593](#)
-
- checksum, 532-534
-
- DNS,
- [615-616](#)
-
- mechanismo de controle interno, 532-534
-
- multicasting e broadcasting, 532-534
-
- para comunicação simplificada, 532-534
-
- portas, 787
-
- programa cliente, 791-794
-
- programa servidor, 780
-
- protocolos de atualização de rotas, 532-534
-
- RTP, 679
-
- sem conexão, 532-533
-
- SIP,
- [680-681](#)
-
- Copyrighted material

- tráfego em tempo real, 678-679
vantagens, 532-533
- U-frame*, 266-267, 270
campo controle, 268-269
códigos, 268-269
comandos e respostas, 270
função, 268-269
HDLC, 268-269
FPI, 278
sistema de gerenciamento, 266-267
tipos, 268-269
- UHF, 186-187
- UNI, 403-404
tamanho VPI, 405-406
- Unicasting*, interface do roteador, 483-484
- Unicode*, 35
- UNIs, 405-406
- Unnumbered frame*; *veja U-frame*
- Uplink*, 381-382
- Urgência de dados, 553
- URL
alias, 642-643
componentes, 642-643
documento dinâmico, 653-654
host, 642-643
HTTP, 642-643
locators, 642-643
método, 642-643
nome do caminho, 642-643
número da porta, 642-643
recuperando documentos, 642-643
- User Datagram Protocol; *veja UDP*
- User Mobile Link (UML), 384
- User Network Interface; *veja UNIs*
- UTP; *veja par trançado*
- UU, 412-413
- UUI, 410-411
- V**
- V.32, 142-144
V.32 bis, 142-144
V.34, 143-144
V.90, 143-144
V.92, 144-145
Valor AC, 679
Variável de controle, 254-255
VC, 403-405
VCI, 392, 405-406
comutador VPC, 405-406
tamanho, 405-406
- VDSL, 216-217
- Velocidade de propagação, 94-95
comprimento de onda, 95-96
- Verificador
cabecalho, 465-466
campo, 242
checksum, 235-236, 242-243, 548-549, 768
complemento, 242, 769, 770
CRC, 240-241
erros não detectáveis, 243-244
exemplo, 243-244, 465-466
gerador, 242
hexadecimal, 769
performance, 243-244
procedimento, 242
procedimento do receptor, 243
procedimento do transmissor, 243
receptor, 243
rejeição no receptor, 243-244
soma parcial, 769, 769-770
sum, 780-782
UDP, 532-534
valor no receptor, 243-244
verificador, 243
- Very high bit rate Digital Subscriber Line; *veja VDSL*
- Very Low Frequency; *veja VLF*
- VHF, 186-187
- Video, 35-36, 470-471, 666
compressão, 667-668
- Videoconferência, 674-675
- Virtual Private Network; *veja VPN*
- VLAN
802.1Q, 364
agrupamento através de características múltiplas, 363
agrupamento através de endereços IP, 363
agrupamento através do número de porta, 363
características dos membros, 361-362
comunicação entre switches, 364
conceito, 360-361
configuração, 363
configuração automática, 363
configuração manual, 363
configuração semi-automática, 364
domínio de broadcast, 361-362
endereço IP multicast, 363
frame tagging, 364
LAN lógica, 361-362
manutenção da tabela, 364
TDM, 364
vantagens, 364
- VLF, 186-187
- VoFR, 400
- Voice over Frame Relay; *veja VoFR*
- Voice over IP, 685, 689-690
- Voz
taxa de amostragem, 666
transmissão, 175
VoFR, 400
VP, 403-405
VPI, 405-406
VPN, 742-743
- VRC, 237-238
confiabilidade, 236-237
CRC, 238-239
- VT, 749
- W**
- WAN, 43-44
- WATS, 205-206
- W-CDMA, 379-380
- WCF, 576
- WDM, 155-156, 160
- Web site, 648-649
- WFQ, 569-570
- Wide Area Network; *veja WAN*
- Wide Area Telephone Service; *veja WATS*
- Wireless, 329
camada física, 330-331
campo controle do frame, 335-336
CSMA/CA, 333-335
CSMA/CD, 333-335
frame da camada MAC, 335-336
frame de controle, 335-336
frame de dados, 336-337
mechanismo de endereçamento, 336-337
NAV, 335-336
tipos de frames, 335-336
- WLAN, 329
- World Wide Web; *veja WWW*
- WWW, 642-643, 647-648
conceito, 648-649
distribuição de informação, 649
documento ativo, 655
documento dinâmico, 653-654
documento estático, 649
hipertexto e hipermídia, 648-649
homepage, 648-649
ponteiros, 648-649
tipos de documentos, 649
- X**
- X.25, 396-397
- X.509, 724-725
- xDSL, 213-214
- Z**
- Zona, 606-60

BEHROUZ A. FOROUZAN

COMUNICAÇÃO DE DADOS E REDES DE COMPUTADORES

Obra que oferece uma excelente e elaborada exposição pedagógica sobre o tema. Sua abordagem prática sobre as características de diversos dispositivos utilizados nas redes de comunicações com computadores torna a leitura desta obra indispensável. *Comunicação de Dados e Redes de Computadores*, 3.ed., está organizado de acordo com o modelo das cinco camadas da Internet e incorpora os recentes avanços no campo das redes de computadores. Apresenta ainda exemplos resolvidos, aplicações do dia a dia, numerosos problemas teóricos e práticos, e um glossário com mais de mil termos nas áreas de Comunicação de Dados, Redes de Computadores e Internet.

Esta edição inclui:

- LANs sem fio (*wireless LANs*) e Bluetooth
- Redes de satélites e de telefonia celular
- Material sobre o acesso digital de alta velocidade (a Internet)
 - DSL
 - TV a cabo
 - SONET
- Profundidade nos protocolos de acesso ponto a ponto e acesso múltiplo
- Abordagem do protocolo IP e a inclusão do IPv6
- Material sobre UDP e TCP
- Questões relacionadas à segurança de rede
- Material sobre redes sem fios (*wireless*)

A leitura de *Comunicação de Dados e Redes de Computadores*, 3.ed., interessa àqueles que estão cursando Ciência da Computação, Engenharia de Telecomunicações, cursos de especialização em Redes de Computadores e Internet e, também, a todos os profissionais que desejam se atualizar nessa área essencial.

ISBN 978-85-363-0614-8



artmed®
EDITORIA
RESPEITO PELO CONHECIMENTO



www.bookman.com.br

Copyrighted material